PUBLISHED BY SCHOLASTIC INC.

FEBRUARY
1984

VOLUME 2
NUMBER 2

FAMILY COMPUTING™

$1.95

Learning Logo Is a Family Affair

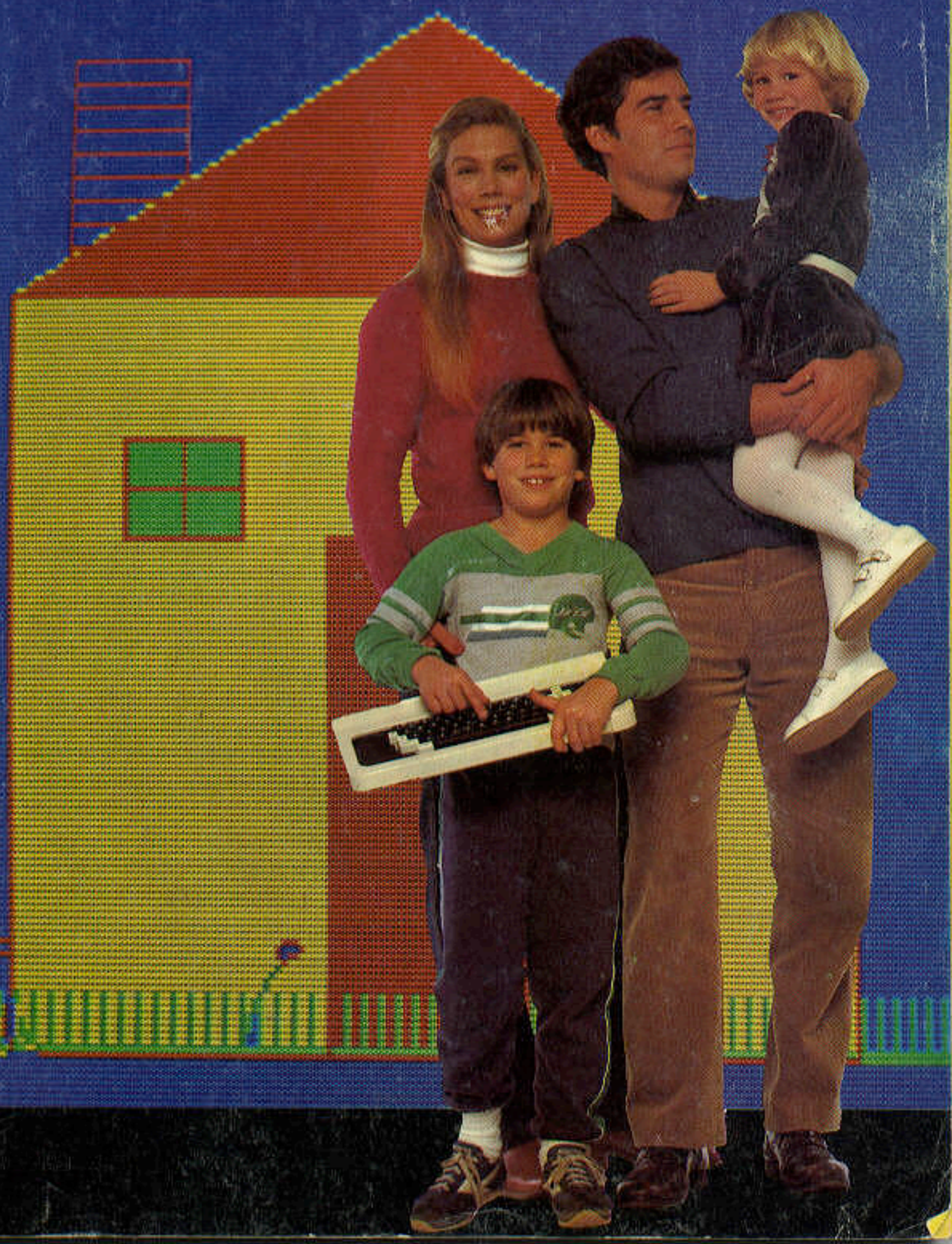
How to Make Sure Your Computer Is Safe

A Mom and Data-Base Management

Don Bluth: Animation and the Games Revolution

More Original Programs for Major Computer Brands

Buyers' Guide to Graphics Tablets, Light Pens and Software



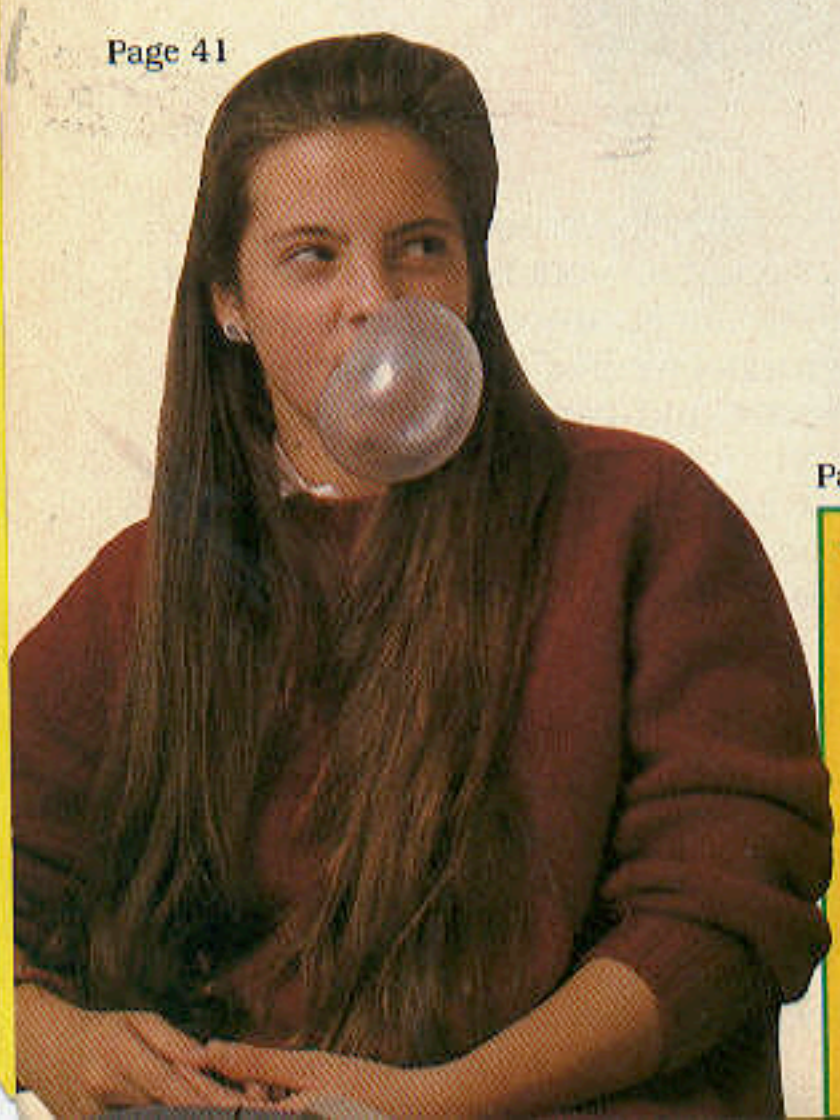
0 76956 14008 02

FAMILY COMPUTING™

FEATURES

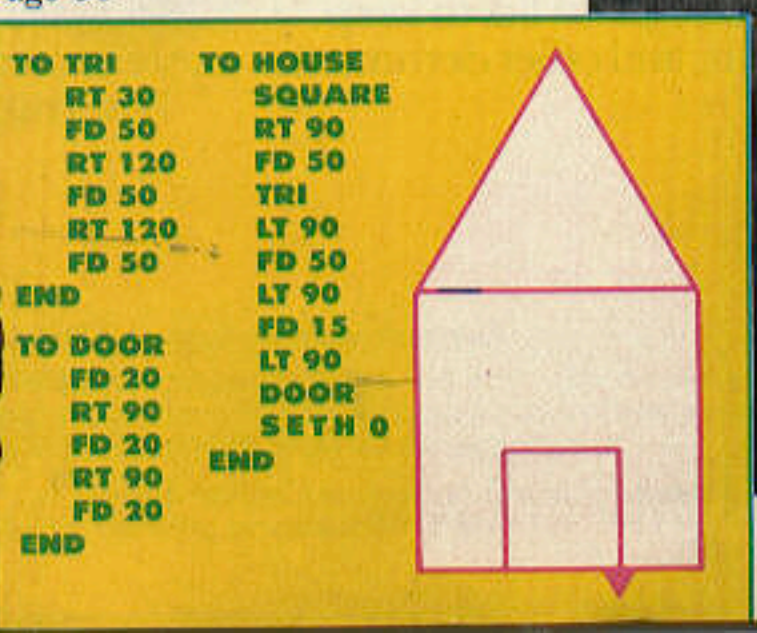
COVER: COMPUTER GRAPHICS CREATED BY JOHN JAINSCHIGG USING LOGO! PHOTOGRAPH BY ANTHONY LOEW.

Page 41

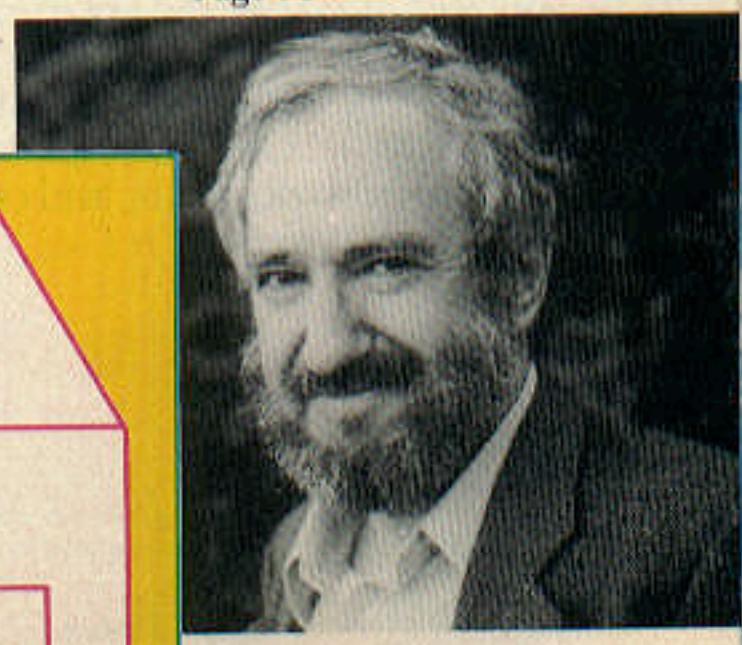


Page 64

```
TO TRI
   RT 30
   FD 50
   RT 120
   FD 50
   RT 120
   FD 50
END

TO DOOR
   FD 20
   RT 90
   FD 20
   RT 90
   FD 20
END

TO HOUSE
   SQUARE
   RT 90
   FD 50
   TRI
   LT 90
   FD 50
   LT 90
   FD 15
   LT 90
   DOOR
   SETH 0
END
```



Page 70



FEBRUARY
1984

PUBLISHED BY SCHOLASTIC INC.

VOLUME 2
NUMBER 2

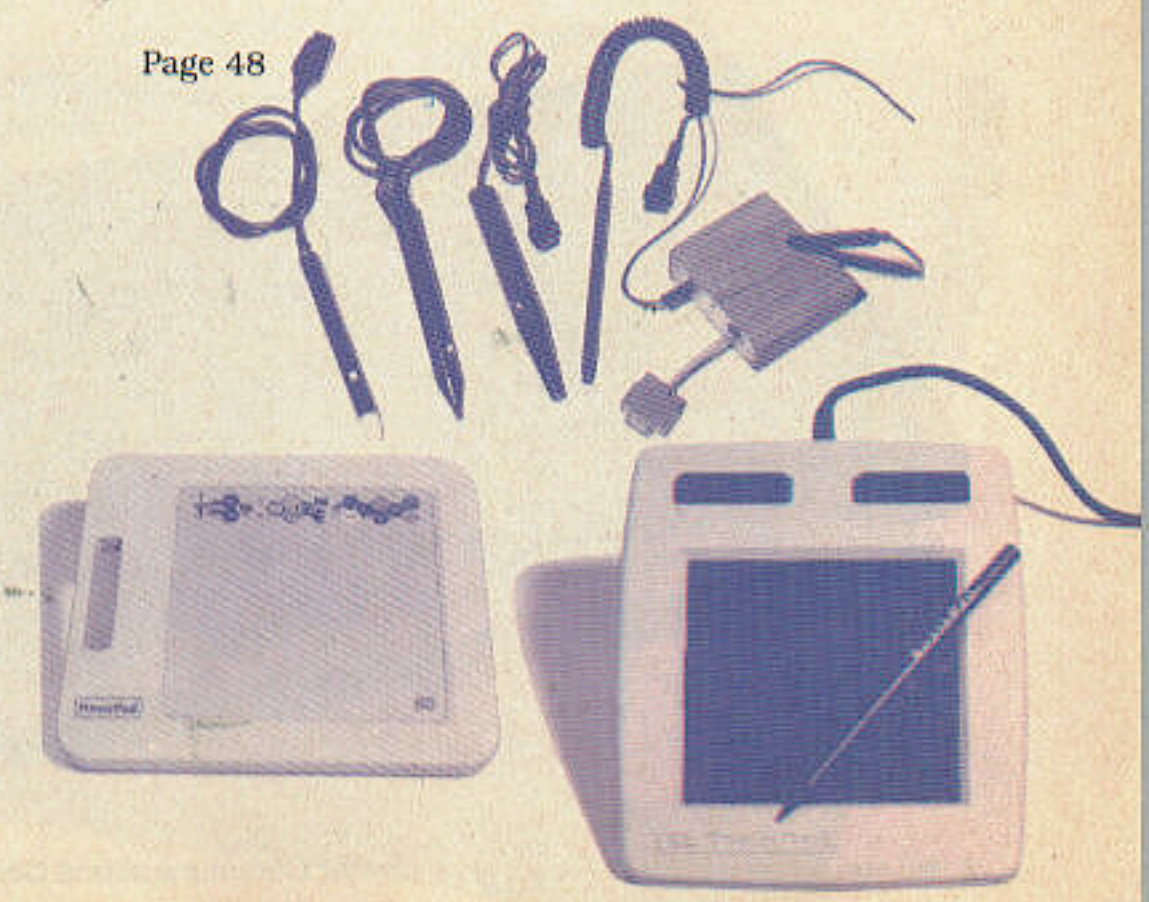
PROGRAMMING

PRODUCTS

DEPARTMENTS

FAMILY COMPUTING (ISSN 0738-6079) is published monthly by Scholastic Inc., 730 Broadway, NY, NY 10003. Subscriptions: in the U.S. and possessions, 12 issues for $17.97; outside the U.S. add $6 (surface mail) and $24 (air mail). Office of publication, Box 2700, Monroe, Ohio, 45050-2700. Application to Mail at Second Class Postage Rates is pending at Monroe, Ohio, 45050-9998 and additional offices. POSTMASTER: Send address changes and notice of undelivered copies to FAMILY COMPUTING, P.O. Box 2511, Boulder, Colorado 80322. Printed in U.S.A.

Page 48



Page 51

EDITOR'S NOTE

WHO'S THIS RELATIONSHIP WITH, ANYWAY?

There are people, I know, who rely solely upon themselves. I'm not one of them. All my life I've been "hooked" on other people. It's never been possible for me to go my own way and just not care. Instead, I've always found myself emboldened by the examples set for me by people I admire. Their independence, self-confidence, and courage gave rise to my own.

As a child I had my assortment of favorite relatives and neighbors, of course, mostly ones who inspired me to follow new dreams. At school there were teachers and later professors whose love for their subjects, for learning, and for life gave me a thirst for knowledge that extended far beyond the classroom. When I went to work, the same was true. I'd work tirelessly, striving always to do better, and it seemed effortless if my boss was someone I admired. These people made a difference in every aspect of my life.

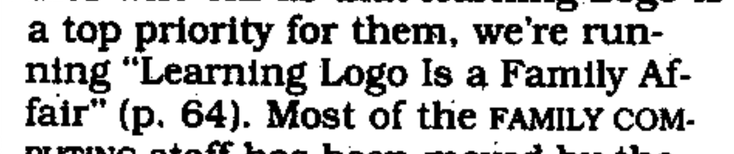
A connection to people is at the heart of our goals for FAMILY COMPUTING. It brings us sheer pleasure to forge a new path in an industry that's been driven by machines. Detractors are out there, fighting to keep all discussions of computers centered on the machine. Putting the emphasis on people, some say, will never work. "A lousy idea," I've heard some state.

Committed to meeting the needs of our readers, we've forged ahead. Now the response is pouring in, and to open our mailbox is to find a daily batch of letters that say, "Thank you for FAMILY COMPUTING." These are our Valentines—any month they arrive. Other letters request information or help. We regard these as bills—a debt we owe our readers for the trust that is being placed in us.

We try to earn your trust in a number of ways: by publishing articles that help you make better use of your computer, by serving as your advocate when we review new products, by providing you with original programs for every holiday and season as well as year-long fun, and by keeping before us the goal of helping you to fit your computer into your everyday life.

Among the ways in which we're meeting these goals this issue is the commitment we're making in our Programmer section to continue to run TI programs every month for the forseeable future despite TI's announcement that they will no longer produce computers for the home market. And for the countless families who tell us that learning Logo is a top priority for them, we're running "Learning Logo Is a Family Affair" (p. 64). Most of the FAMILY COMPUTING staff has been moved by the story of Michela Alioto ("64 Inches of Courage," p. 41), and we think you'll find this extraordinary teenager equally inspiring.

We always want to do more and better for you, motivated in part by the energy and devotion of so many of our readers. Let us know how we can be more useful and valuable to you. M.R. Robinson, the founder of Scholastic, who died early in February of 1982, often reminded us that being invited into the lives of our readers is a privilege.



CLAUDIA COHL
EDITOR-IN-CHIEF

FAMILY COMPUTING™

FAMILY COMPUTING (ISSN 0738-6079) is published monthly by Scholastic Inc., 730 Broadway, New York, NY 10003. Subscriptions: in the U.S. and possessions, 12 issues for $17.97; outside the U.S. add $6 (surface mail) and $24 (air mail). Office of publication, FAMILY COMPUTING, Box 2700, 351 Garver Rd., Monroe, OH, 45050-2700. Application to mail at Second Class Postage Rates is pending at Monroe, OH, 45050-9998 and additional offices. POSTMASTER: Send address changes and notice of undelivered copies to FAMILY COMPUTING, P.O. Box 2511, Boulder, CO 80322.

LETTERS

READERS OFFER TRANSLATIONS

I read with interest the article on the *Home Heat Loss Calculator* program (Premier issue, p. 85). The TI-99/4A program worked fine on my sister's TI-99, but I am a TS 1000 owner, and I have noted the great majority of interesting programs are not for the TS 1000.

So, I decided to translate the *Heat Loss* program into Sinclair BASIC. The program runs fairly quickly and is serving this household in determining which options for weatherization we will pursue. The program was written on a TS 1000 with a 16K RAM expansion and recorded on a Sanyo ST-45 recorder. I hope you will want to share this translation with your readers who are TS 1000 owners.

VAUGHN R. MARTENS
Racine, WI

After reading your first issue, I like it. So far I have only one complaint: You have nothing for the TRS-80 PC-2. I have to admit that none of the other magazines do. But you claim to include what they do not. Please accommodate those of us who do not have a full-size computer. I altered one of your programs (*Future Age Calculator*) for my PC-2 so it would run. But, if I was a great programmer, I would not need your magazine.

WALLACE E. WILLIAMS
Interlaken, NY

EDITOR'S NOTE: *Your letter points out the need for us to be more specific about which models our programs run on. Among the TRS-80 machines, we run programs for Models I, III, 4, and the Color Computer—the most popular brands for family use. Unfortunately, your machine does not fit that category.*

Although FAMILY COMPUTING *would like to publish all reader-written program translations, it is impossible because of time and space limitations. For Timex users who would like a copy of the adapted program for Home Heat Loss, send a self-addressed, stamped envelope to Mr. Martens, 3504 Six Mile Rd., Racine, WI 53402. TRS-80 PC-2 users who would like a translation of the Future Age Calculator program, send a self-addressed, stamped envelope to Mr. Williams, RD #2, Box 253, Potter Rd., Interlaken, NY 14847.*

FAMILY COMPUTING *has not had the opportunity to test these programs properly; therefore, we cannot bear responsibility for any damage to your equipment that may result.*

DRACULA'S PROGRAM: A BAD BLOODLINE?

I was wondering to what extent the programs that appear in your magazine are tested before they are printed in each issue. I am specifically referring to the *Dracula's Family Tree* program in the October issue. I have attempted to run this program several times on a TI-99/4A, using the TI translation. Each time, I get an error statement that reads: INCORRECT STATEMENT IN LINE 210. I am getting thoroughly disgusted with trying to determine why the program won't run and have concluded that it is a programming error.

I would appreciate your verifying if the program is correct, or letting me know what is incorrect about line 210. Your assistance will be greatly appreciated, as you very well may save my sanity.

PAUL L. WILLIAMS
Ashland City, TN

EDITOR'S NOTE: *We test our programs extensively; unfortunately, we occasionally make errors. There was an error in the modification box for the Texas Instruments version of* Dracula's Family Tree. *We apologize for any inconvenience. The corrected version was printed in the December issue, page 143.*

A PEACH OF A PUMPKIN

Just completed your most colorful pumpkin on a C-64. Beautiful! Come the end of the month the monitor in the window should make quite an impression on the neighbors. For once the pumpkin won't end up in the street.

Between now and then I want to get a regular subscription to FAMILY COMPUTING, including the first issue and any subsequent issues until the computer-mechanical-process gets my name on your mailing list. I don't want to miss even a single edition.

In a world where new computer publications are appearing almost daily, yours is TOPS. Keep up the good work.

RUSSELL S. DECHSLIN
Spencer, IA

DESIGNING A DATA BASE

Just finished my first FAMILY COMPUTING. Super is about the best word.

I enjoyed your different programs. I had a few questions answered by studying these.

Do you have, or are you going to run, a program that designs a filing system? I teach, and am looking for a file program to store information with cross-reference and search capabilities. I would like to program one myself.

Thanks for a fine publication.

LARRY THOMAS
Hooks, TX

EDITOR'S NOTE: *We are featuring some examples of commercial data-base-management programs in this issue, page 78. Unfortunately, we have no stories scheduled on how to program your own data base.*

A PLEA FOR NO TREE

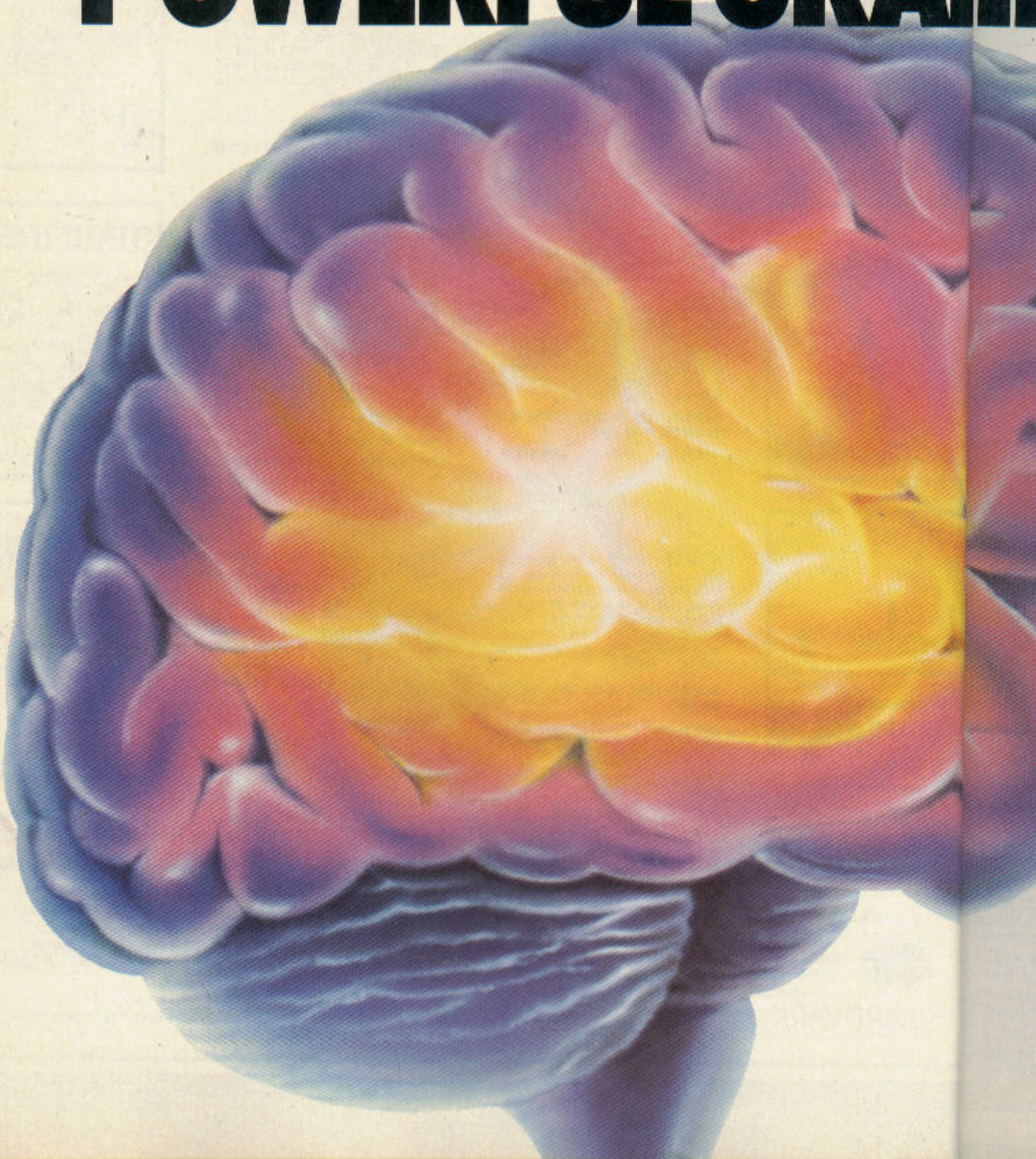
When my husband suggested we use your *Jack-O'-Lantern* (October issue) program instead of our traditional pumpkin carving, we all agreed it was a great idea. Our three-and-a-half-year-old, Kenneth, enjoys "carving" his own pumpkin by pushing RUN! But, please—in your Christmas issue, don't include a Christmas tree! My stepfather sells Christmas trees, and I'd be kicked out of the family if we didn't have a bushel of pine needles to clean up the first week of January! And, Kenny doesn't think we could fit enough gifts under your version.

MARGUERITE KISTLER
Perkasie, PA

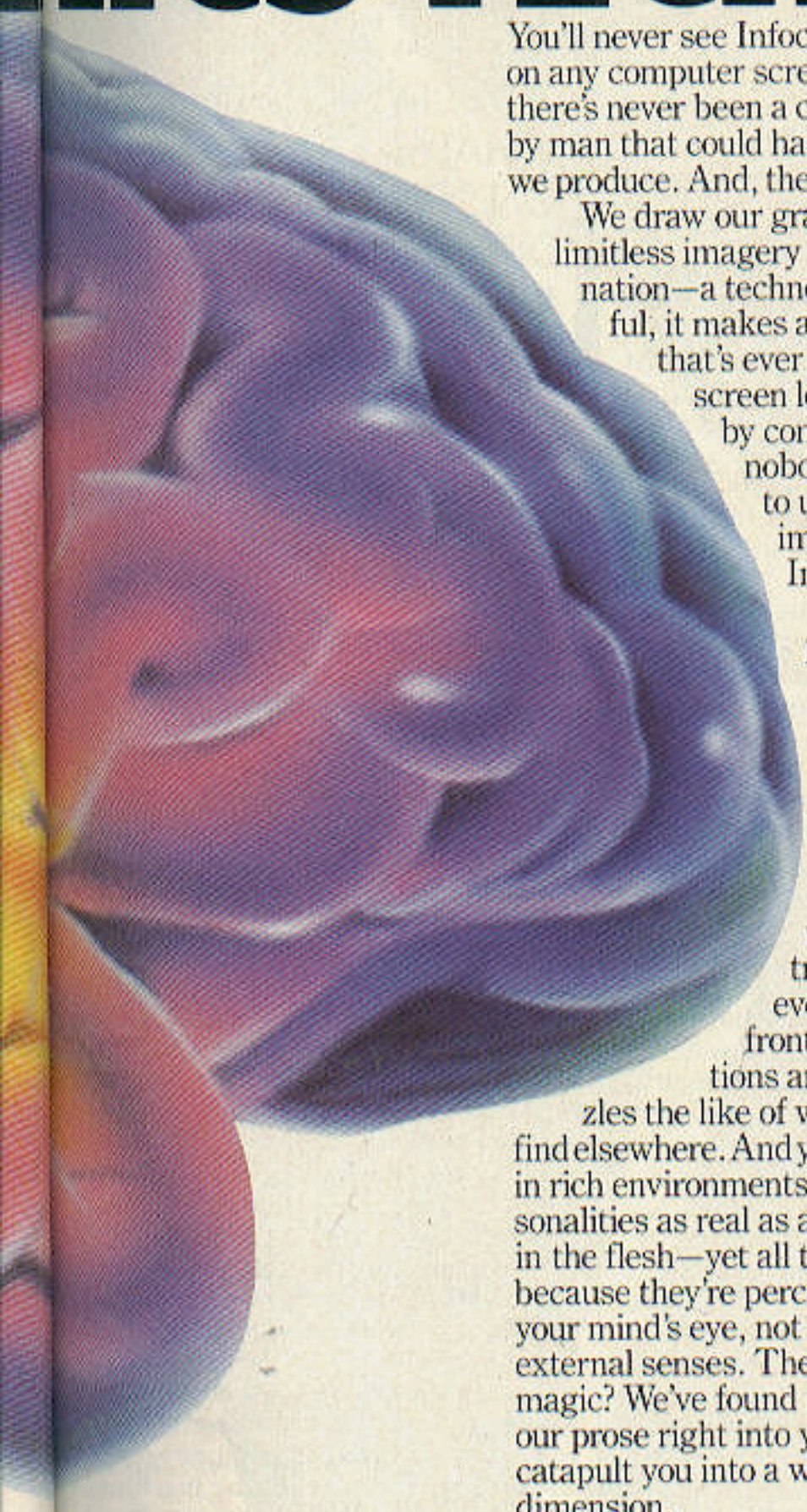
EDITOR'S NOTE: *Sorry, it's too late; our* Christmas Tree *program appeared on page 110 of the December issue. We hope it was the "extra" tree and extra present we intended it to be.*

FAMILY COMPUTING *looks forward to letters from all our readers. Please direct your correspondence to: Letters to the Editor,* FAMILY COMPUTING, *730 Broadway, New York, NY 10003. Include your name, address, and phone number. We reserve the right to edit your letters for length and clarity.*

WE UNLEASH TE
POWERFUL GRAH



E WORLD'S MOST HICS TECHNOLOGY.



You'll never see Infocom's graphics on any computer screen. Because there's never been a computer built by man that could handle the images we produce. And, there never will be.

We draw our graphics from the limitless imagery of your imagination—a technology so powerful, it makes any picture that's ever come out of a screen look like graffiti by comparison. And nobody knows how to unleash your imagination like Infocom.

Through our prose, your imagination makes you part of our stories, in control of what you do and where you go—yet unable to predict or control the course of events. You're confronted with situations and logical puzzles the like of which you won't find elsewhere. And you're immersed in rich environments alive with personalities as real as any you'll meet in the flesh—yet all the more vivid because they're perceived directly by your mind's eye, not through your external senses. The method to this magic? We've found the way to plug our prose right into your psyche, and catapult you into a whole new dimension.

Take some tough critics' words about our words. SOFTALK, for example, called ZORK® III's prose "far more graphic than any depiction yet achieved by an adventure with graphics." And the NEW YORK TIMES saw fit to print that our DEADLINE™ is "an amazing feat of programming." Even a journal as video-oriented as ELECTRONIC GAMES found Infocom prose to be such an eye-opener, they named one of our games their Best Adventure of 1983.

Better still, bring an Infocom game home with you. Discover firsthand why thousands upon thousands of discriminating game players keep turning everything we write into instantaneous bestsellers.

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

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

INFOCOM™

The next dimension.

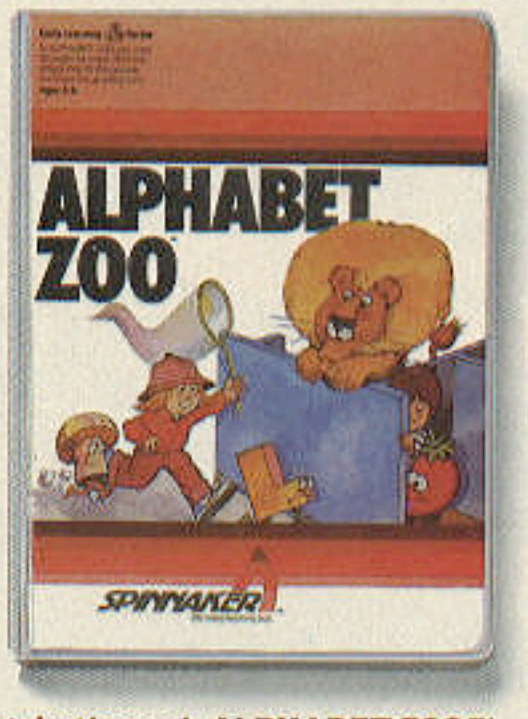
For your: Apple II, Atari, Commodore 64, CP/M 8," DEC Rainbow, DEC RT-11, IBM, MS-DOS 2.0, NEC APC, NEC PC-8000, Osborne, TI Professional, TI 99/4A, TRS-80 Model I, TRS-80 Model III.

T PARENTS, YOU WON'T
E SPINNAKER GAME.

FRACTION FEVER™ brings fractions into play.
Ages 7 to Adult.

FRACTION FEVER is a fast-paced arcade game that challenges a child's understanding of fractions. As kids race across the screen in search of the assigned fraction, they're actually learning what a fraction is and about relationships between fractions.

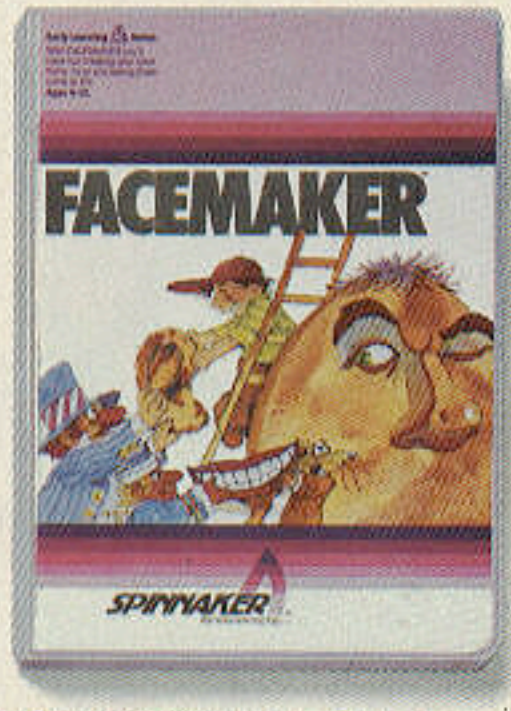
All in all, FRACTION FEVER encourages kids to learn as much as they can about fractions—just for the fun of it!



A trip through ALPHABET ZOO.™
Ages 3 to 8.

It's a race. It's a chase. It's Alphabet Zoo, a game that sends your kids zipping through the maze, after letters that fit the picture on the screen.

Your kids will have fun learning the relationship of letters and sounds, and sharpening their spelling skills. They'll be laughing at every turn.



FACEMAKER™ makes faces fun.
Ages 3 to 8.

FACEMAKER lets children create their own funny faces on the screen, then make them do all kinds of neat things: wink, smile, wiggle their ears, and more.

Plus, FACEMAKER helps familiarize children with such computer fundamentals as menus, cursors, simple programs, and graphics. FACEMAKER won't make parents frown because their children will have fun making friends with the computer.

SPINNAKER™
We make learning fun.

Disks for: Apple, Atari, IBM PC and PCjr, Commodore 64.
Cartridges for: Atari, IBM PCjr, Commodore 64, Coleco Adam.

A Novices' Guide to Programming LaNgUAGes

TO MAKE A COMPUTER WORK
YOU NEED A PROGRAM.
TO WRITE A PROGRAM YOU NEED TO
KNOW A LANGUAGE.

BY KENNETH P. GOLDBERG

If you own a microcomputer, there are three basic ways you can use it. You can buy and use ready-made software—prerecorded programs; you can write your own programs; or you can do a little of both. If you plan to do some programming of your own, either because you can't find commercial software that does exactly what you want or for the challenge and enjoyment of it, then you'll need a programming language that is specifically suited to your purposes.

A programming language, just like a spoken language, is a set of rules that allows you to communicate—in this case, with a computer. For the computer to understand the language you use—be it BASIC, Pascal, Logo, etc.—that language must first be loaded into the computer's memory.

Most microcomputers come with the BASIC language built in and will understand BASIC programs. To write and run a program in another language, you must purchase that language and load it into the computer's RAM. These languages usually come in the form of a cartridge, card, or circuit board, which fits into a slot in the computer, or on a disk, which is loaded into the computer like any other program.

Following are descriptions of several commonly used programming languages that are available for microcomputers, along with brief explanations of how they work and what they are best suited for.

KENNETH P. GOLDBERG *is chairperson of the Math, Science and Statistics Education department at New York University. He is the author of* Microcomputers: A Parents' Guide *(John Wiley & Sons) and* The Parents' Book on Calculators *(Oxford University Press).*

—A beginners' general-purpose language

BASIC

```
10  HOME
20  FOR A = 1 TO 1000000
30  PRINT A;
40  NEXT A
50  HOME
60  PRINT "ALL DONE!"
```

This BASIC program counts from 1 to 1,000,000 (displaying all the numbers on the screen), and then prints ALL DONE!

ASIC (Beginner's All-purpose Symbolic uction Code) was developed at Dartmouth e in the mid-1960s to allow people who not computer specialists to make use of ower of the computer. Whereas it might taken a programmer months or even to become proficient in the computer lan- s that were available before the develop- of BASIC, a nonspecialist can probably useful programs in BASIC after only a few of practice, and fairly sophisticated ones only a few weeks or months.

or those who quickly master BASIC, some uters (TRS-80 Color Computer and TI-99/ fer Extended BASIC, which is slightly powerful and allows you to write pro- with sophisticated graphic displays.

ASIC has a relatively small number of ands (statements that tell the computer cute certain steps), most of which are milar to English words and statements. they are easy to learn and remember. To stand BASIC's commands, the computer n internal translator that turns BASIC he computer's own language, or machine age.

his translator is called an interpreter. In- ters can be fairly slow, so people who their programs to be executed faster often compiler. This translates BASIC com- into machine language more quickly.

en though BASIC sounds like English, an't use just any English words or s, but only those that the "phrase book" ted. For example, you can tell the com- puter to display the word *computer* on its monitor using the command PRINT "COMPUTER", since PRINT is an acceptable command. But you can't do it by typing WRITE THE WORD COMPUTER ON THE MONITOR, because that command is not in the phrase book.

While BASIC has never been thought of as the best language for young children (see Logo for one that is), there is now some question whether BASIC is an appropriate first language for any programmer. The very thing that makes BASIC easy to learn in the first place—its small number of commands—sometimes makes it difficult to write sophisticated programs in a straightforward manner. As a result, many education experts contend that BASIC is difficult to use and teaches bad programming habits that interfere with learning more advanced languages.

For the time being, however, BASIC is still the language that comes with most computers, and for that reason remains the most common first language. In many cases, it is the only language computer users ever learn.

—An educational graphics language

LOGO

```
TO HISTOGRAM :LIST
IF EMPTYP :LIST
[PR [TASK COMPLETE.] STOP]
MAKE "NUM FIRST :LIST
MAKE "LIST BUTFIRST :LIST
TYPE :NUM STARS :NUM PR []
HISTOGRAM :LIST
END

TO STARS :NUM
IF EQUALP :NUM 0 [STOP]
TYPE "*
MAKE "NUM :NUM - 1
STARS :NUM
END
```

This Logo program creates a histogram, a type of graph.

ogo is an education-oriented language specifically at the elementary school–age was invented by Seymour Papert, a at the Massachusetts Institute of in the 1960s. The first microcom- developed for was the TI-99/4, now available for many computers, the Apple, Atari, Commodore, IBM, nstruments, and TRS-80. There are sev- rsions for the Apple and IBM.

he educational philosophy of Logo is to a learning environment, which Papert signated "Mathland," in which young can explore, experiment, and thereby about computers, mathematics, and own thinking processes.

entral to this philosophy is Logo's graph- ability, called "turtle graphics" because iginal version of Logo allowed children to computer to command a mechanical ob- wheels that looked like a turtle. In the t versions of Logo, the turtle is an object computer screen rather than an actual nical object.

he screen turtle can be made to move and y using a simple vocabulary of words—FORWARD, BACKWARD, RIGHT, and LEFT—each followed by a number. The number following FORWARD and BACKWARD tells the turtle how many "turtle steps" to move in either direction, respectively. The number following RIGHT and LEFT tells the turtle how many degrees to turn in either direction. As the turtle moves, it leaves a trail in one of several possible colors, and it is this trail that creates drawings and designs. For example, the command REPEAT 4 [FORWARD 50 RIGHT 90] tells the turtle to move forward 50 steps and turn 90 degrees to the right, and repeat this four times. The drawing created by this command would be a square with sides the length of 50 turtle steps.

Several versions of Logo also include "sprites," invisible "creatures" that can be given different shapes and colors and set in motion to provide realistic and impressive animation. As the children try to "teach" the turtle and/or sprites to move around and draw pictures, they automatically learn about such concepts as direction, distance, motion, shape, and color.

In addition to graphics, Logo also provides arithmetic and list-processing capability. While the computational capability is not as extensive as that provided in BASIC or some other languages, it is more than sufficient for schoolwork, household finance, and simple business applications. List processing allows the user to create, store, and manipulate lists of numbers, words, and sentences.

Logo is adapted from the language LISP (LISt Processing), which has only list-processing capabilities and no graphics. LISP is used in artificial-intelligence research as a language that allows a computer to mimic human behavior. List processing is very useful for this purpose because it allows the computer to be given a store of basic English phrases that it can manipulate and modify as it holds a seemingly human "conversation" with a human being.

List processing does allow for the recording, modification, and manipulation of text material such as names, addresses, and inventory, so versions of Logo developed for a business computer like the IBM PC stress list processing. But for the average computer owner, the chief attraction of Logo is its graphics capability and educational uses rather than its list-processing applications.

Unlike BASIC, Logo is what is called a "modular language," because it allows complicated problems to be broken down into simpler modules, or components. These can then be solved independently of each other and recombined for a solution to the original problem. Pascal, a language that is very popular at colleges, is also a modular language. Because of this, many educators now feel that Logo is a better first computer language for children than BASIC. Logo teaches good programming habits, and the translation from Logo to Pascal is much easier than from BASIC to Pascal.

If education is of primary concern to you, either to allow your children to explore and learn about mathematics, computers, and graphics on their own, or to keep up with the possible use of Logo in their school, Logo may be a useful language for you to own.

PILOT—*A learning and teaching language*

PILOT (Programmed Inquiry, Learning, Or Teaching) is an authoring language designed expressly to allow for the easy creation of computer-aided-instruction (CAI) programs. In these, information is presented and questions are asked; the user responds; and the program branches to one of several alternatives depending on the response given.

Although PILOT was originally designed to be used by educators, its ease of use and excellent graphics capability also make it of possible interest to noneducators who want to develop

This Pilot program sets up a little quiz, asking the user to "name a product exported by Brazil."


```
10   T: TODAY'S TEST QUESTION IS ON BRAZIL
20   T:
30   T: NAME A PRODUCT EXPORTED BY BRAZIL
40   T:
50   A:
60   M: COFFEE RUBBER GEMS
70   T:
80  TY: CORRECT.  VERY WELL DONE
90  TN: SORRY.  BETTER LUCK ON TOMORROW'S QUESTION
```

educational programs for their children or who just want an easy way to create computer games that combine text with graphics.

In PILOT, each program line begins with a code letter (or two) that tells the computer what to do with the rest of the line. For example, "T:" at the beginning of a line tells the computer to "type," or display on the screen the message that follows. Similarly, "C:" tells the computer to "compute" the mathematical expression that follows; and "M:" tells the computer to expect an input from the user and look to see if this input "matches" any of the words that follow.

The "M:" code is especially useful in the authoring of educational programs. For example, suppose you want to ask the user to name one of the products exported by Brazil. You will accept either "coffee," "rubber," or "gems" as being correct. The command M: COFFEE, RUBBER GEMS will look at the user response and accept any *one* of these three words as correct.

This automatic coding of program lines frees the programmer from having to tell the computer *what* to do with each line, so that he or she can concentrate on the *content* and *design* of the program.

PILOT is relatively easy to learn, even for the novice, and allows the user to begin authoring CAI programs almost immediately. One drawback of PILOT is that it executes, or "runs," the program more slowly than some other languages. This is because the computer must "decode" the code letters at the beginning of each line as it runs through the program. However, this is what you give up for the ease of development PILOT offers. And, since the difference in execution speed is only a matter of seconds, it isn't very noticeable.

PILOT is available for several of the more popular computers. The Atari, Apple, and Commodore versions contain turtle-type col-

graphics very similar to those offered in Logo. They can be used alone, or in conjunction with the text and computational capabilities of the language to produce very sophisticated instructional programs.

If you want to create instructional programs, PILOT is specifically designed to help you do this easily and well. But if you have some other application in mind, stick to BASIC or some other language more suited to that task. You may be able to do it with PILOT, but probably not as easily or as well.

PASCAL—*A classic structured language*

Pascal

Pascal, named after the 17th-century French mathematician Blaise Pascal, is a "structured" language. This means that programs in Pascal must follow certain built-in rules of "good programming" or they will not run.

For example, a program in Pascal must begin with the word PROGRAM followed by the name of the program; it must then have the word VAR (for variable) followed by a list of all the names of items that will be used in the program; it must then have the word BEGIN followed by the actual commands of the program; and, finally, it must have the word END to tell the computer the end of the program has been reached.

Pascal, as noted before, is a modular language that allows complicated problems to be broken down into simpler modules. For example, suppose a numerical problem requires a different method of solution depending on whether all the numbers involved are positive, all the numbers involved are negative, or the numbers involved are of both types. Individuals or teams can be assigned to work independently of each other on the three problem types, and their respective solutions can then be assembled to form a complete solution covering all three possibilities. Any of these modules can then later be modified or even removed without affecting any of the others.

```
PROGRAM ADDSUB;
   TYPE MATH =(ADD);
   VAR CONST1, CONST2, RESULT : INTEGER;
       OPERATION : MATH;
   BEGIN
      CONST1 := 5;
      CONST2 := 7;
      OPERATION := ADD;
      IF OPERATION = ADD THEN
      BEGIN
         RESULT := CONST1 + CONST2;
         WRITELN('THE ANSWER OF 5+7 IS ', RESULT)
      END
   END.
```

To add 5 and 7 in BASIC, you merely have to type, PRINT 5 + 7. *To add 5 and 7 in Pascal, you need a lengthier program, such as the one shown here.*

While Pascal is not as easy to learn as BASIC, once it is learned it allows for more powerful and efficient programs, programs that run faster and are easier to modify. Because of these desirable characteristics, Pascal is fast replacing BASIC in many colleges as the first programming language taught and is being

PASCAL CHOSEN FOR AP EXAM

While BASIC is the "native language" of most home and personal computers, as well as the cornerstone of computer literacy, its preeminent position may be in jeopardy. Pascal was the language selected for the first computer science advanced placement (AP) exam, to be given this May. As a result, high schoolers wanting a jump on college computer courses will have to master Pascal.

AP exams allow high school students to earn some college credit for their work in such subjects as English, history, calculus, and music. In participating high schools, students take special courses that are equivalent to introductory college classes, as determined by committees of college and high school instructors. The committees also develop course guides and exams.

Approximately 6,000 high schools are preparing students for the AP exam in computer science this year, according to Anne Grosso, a spokesperson for the College Entrance Examination Board (CEEB), which develops AP exams. As yet, no one knows how many students will actually take the standardized computer science test, but estimates range from 3,000 to 12,000. (Last year 51,000 students took the English literature and composition exam, while only 600 vied for credit in music.) The newest three-hour exam will consist of two parts: one with multiple-choice questions; the other presenting a situation or problem that requires students to write a computer program in Pascal.

Some educators have challenged the decision to base the exam on Pascal, since BASIC is more commonly taught in high schools. In addition, the choice may create a new class of haves and have-nots—those attending schools that offer Pascal instruction and those who don't. Yet the CEEB says the planning committee polled 200 colleges and determined that Pascal is more widely taught than BASIC. Harlan P. Hanson, the CEEB's AP Program Service Officer, says: "It would be dreadful if what [students] were doing in high school had no relation to what followed in college." —LINDA WILLIAMS

adopted for business applications as well. Pascal has also been chosen as the programming language to be used in the advanced placement (AP) examination in computer science for high school students going on to college [*page 57*].

One minor drawback of Pascal is that, unlike BASIC and Logo, it cannot be used to write statements that will be carried out immediately by the computer. For example, in BASIC, the command PRINT "BLAISE PASCAL" will cause the computer to immediately display the name BLAISE PASCAL on its monitor as soon as the command is entered. No such immediate execution is possible in Pascal. Instead, you would have to write a program commanding the computer to print the name and then tell the computer to run the program.

This is a very minor drawback, however, since the vast majority of applications to which a computer language is put involve programs that will be used over and over again, rather than for the immediate evaluation of a mathematical expression or the immediate printing of a message on the display screen.

Pascal might be a good language to own if you have a college student in your family or a high school student who plans to go to college. It is also the right language for anyone who intends to do serious programming and wants the advantages Pascal offers over BASIC in programming efficiency, speed of execution, and ease of modification.

MACHINE LANGUAGE—*The computer's native language*

There are easier ways to add 5 and 7 than writing a program in machine or assembly language. But we did it anyway, just to show what the languages look like.

Machine language lets you communicate directly with the computer. It is the language the computer understands. The commands are composed of binary digits, i.e., strings of 0s and 1s. Machine language uses very little memory space, gives you total control of the color and sound capabilities of your computer, and can be executed very fast. This is because the computer does not have to internally translate your commands, as it must with so-called high-level English-like languages—BASIC, Logo, PILOT, and Pascal.

MACHINE

```
0010000001011000111111000001100011111000101010010000010101101001000001111000110100100111000000111101100001000101001010010010100100101000101001000011110110100110000000100000111011011111111010111010011001011000000100000111011011111111010010000011010000000000011000000000
```

If, for example, you write a program telling the computer to count to one million and then display the message I'M DONE, a program written in machine language would do this at least a thousand times faster than an equivalent BASIC program. For this reason, many fast-paced action games are written in machine language.

In spite of its speed of execution, however, machine language is relatively difficult to learn. And it's not hard to make mistakes, because the programmer must tell the computer exactly what to do every step of the way, with no shortcuts from a built-in "phrase book."

For instance, to write a program in machine language to add 5 and 7, you have to tell the machine 1) to put the first number, 5, in a specific memory location; 2) to put the second number, 7, in another memory location; 3) to take each of the numbers out of their memory locations and put them in a special computation location; 4) to perform the operation of addition and place the result in a specific memory location; and finally 5) to take the sum from its memory location and display it on the screen.

In BASIC, by comparison, all you need to do is give the command PRINT 5+7, and the computer automatically does everything else. Machine language is certainly a language for the advanced programmer.

ASSEMBLY—*The user-friendly "machine" language*

ASSEMBLY

```
0300-  20 58 FC   JSR $FC58  ;CLEAR THE SCREEN.
0303-  18         CLC        ;CLEAR CARRY FLAG.
0304-  F8         SED        ;SET DECIMAL MODE.
0305-  A9 05      LDA #$05   ;LOAD 5 INTO ACCUMULATOR.
0307-  69 07      ADC #$07   ;ADD 7 TO ACCUMULATOR.
0309-  8D 27 03   STA $0327  ;STORE THE RESULT IN LOCATION 327.
030C-  D8         CLD        ;CLEAR DECIMAL MODE.
030D-  4A         LSR A      ;SHIFT RIGHT ONE BIT.
030E-  4A         LSR A      ;SHIFT RIGHT ONE BIT.
030F-  4A         LSR A      ;SHIFT RIGHT ONE BIT.
0310-  4A         LSR A      ;SHIFT RIGHT ONE BIT.
0311-  29 0F      AND #$0F   ;"AND" MEMORY WITH ACCUMULATOR.
0313-  69 B0      ADC #$B0   ;CONVERT TO ASCII.
0315-  20 ED FD   JSR $FDED  ;OUTPUT (ASCII) CHARACTER TO SCREEN.
0318-  AD 27 03   LDA $0327  ;LOAD INTO ACCUMULATOR VALUE STORED
                             IN LOCATION 327.
031B-  29 0F      AND #$0F   ;"AND" MEMORY WITH ACCUMULATOR.
031D-  18         CLC        ;CLEAR CARRY FLAG.
031E-  69 30      ADC #$B0   ;CONVERT TO ASCII.
0320-  20 ED FD   JSR $FDED  ;OUTPUT (ASCII) CHARACTER TO SCREEN.
0323-  20 D0 03   JSR $03D0  ;GO BACK TO BASIC.
0326-  00         BRK        ;END OF PROGRAM.
```

Assembly language lies somewhere between machine language and the high-level languages. It does everything machine language does, but you can use English words and abbreviations to stand for machine instructions and memory locations, instead of figuring out the strings of 0s and 1s. Again, many games are written in assembly language.

Assembly uses mnemonics that hint at what the commands mean, such as CLR to "clear," or empty, a specified memory location; and SWP to "swap," or switch, the contents of two specified memory locations. Because of these features, many nonprofessional programmers choose assembly language over machine language.

FORTH—*A "mid-level" language*

FORTH was developed in the mid-1970s as a way of using the computer to run other machinery, and for business applications with the exception of large-scale numerical computation. It is one step closer to the high-level languages than assembly, with some of the benefits of each. FORTH is becoming much more popular and might be worth looking into if you want a programming language that is faster in execution and more flexible in programming capability than Pascal and BASIC—but not as difficult to learn, or as easy to err with, as assembly.

Interestingly, a recently announced low-priced computer (list price is approximately $150) called the Jupiter Ace has the FORTH language built into it instead of BASIC. This is an interesting development and might herald the appearance of computers with a language other than BASIC as a standard feature.


```
0 ( PRINTS HISTOGRAM OF VALUES ON STACK)
1 ( INITIALIZE ::)
2 0 VARIABLE SP0
3 SP@ SP0 !
4 : DEPTH SP0 @ SP@ - 2 / 1 - ;
5 : STAR 42 EMIT ;
6 : STARZ CR DUP . 0 DO STAR LOOP ;
7 : HIST DEPTH 0 DO STARZ LOOP ;
```

This FORTH program, like the Logo program shown earlier, also creates a type of graph.

FORTRAN—*A scientific language*

FORTRAN (FORmula TRANslator) is a scientific and engineering language originally designed to simplify the programming of algebraic formulas and other mathematical operations. It is extremely computation-oriented. Much of FORTRAN is similar to BASIC, since BASIC is essentially a simplification of FORTRAN for the nonprofessional. However, FORTRAN makes more demands on the programmer than BASIC by requiring such information as whether the numbers being used are integer (5) or "floating point" decimal (5.0); how many places each has after the decimal point if they are decimal; and the form in which numerical results are to be returned. This gives the mathematical programmer more flexibility and choice, but is an added burden for the ordinary programmer. Such information is not required in BASIC, Logo, or PILOT; with a few exceptions, it's all taken care of automatically. FORTRAN is a powerful language for the specialist in mathematical programming, but not really very useful for the average computer user.


```
      LET COUNTER = 0
      DO 10 COUNTER = 1,100
         PRINT,COUNTER
10    CONTINUE
      STOP
      END
```

COBOL—*A business language*

COBOL (COmmon Business-Oriented Language) is a business language developed in large part through the efforts of Grace Murray Hopper of the Department of Defense. It is strong in areas important to business applications, such as large-scale data manipulation, the generation of reports, and the input and output of information, but it is weak in complex mathematical computations (where FORTRAN is strong). COBOL is a well-established language for large-scale business applications, but, again, not desirable for the novice.


```
IDENTIFICATION DIVISION.
PROGRAM-ID.  'DEMO'.
ENVIRONMENT DIVISION.
DATA DIVISION.
PROCEDURE DIVISION.
    PERFORM
    INITIALIZE-COUNTER.
    PERFORM DISPLAY-PARAGRAPH
    THRU COUNT-PARAGRAPH-EXIT
    UNTIL COUNTER > 100.
    STOP RUN.
INITIALIZE-COUNTER.
    MOVE 0 TO COUNTER.
DISPLAY-PARAGRAPH.
    DISPLAY COUNTER.
COUNT-PARAGRAPH.
    COMPUTE COUNTER =
    COUNTER+1
COUNT-PARAGRAPH-EXIT.
EXIT.
```

Both the FORTRAN and COBOL programs shown here count from 1 to 100. Unlike a BASIC program, which puts the results right onto the computer screen, you won't see the countdown unless you enter a command asking for a screen display.

AVAILABILITY

Of these languages, the most widely available for popular microcomputers, besides BASIC, is Logo. For all micros except the Timex, you can find at least one version, ranging in price from $50 to $200. And, soon, there will be a version for Timex as well.

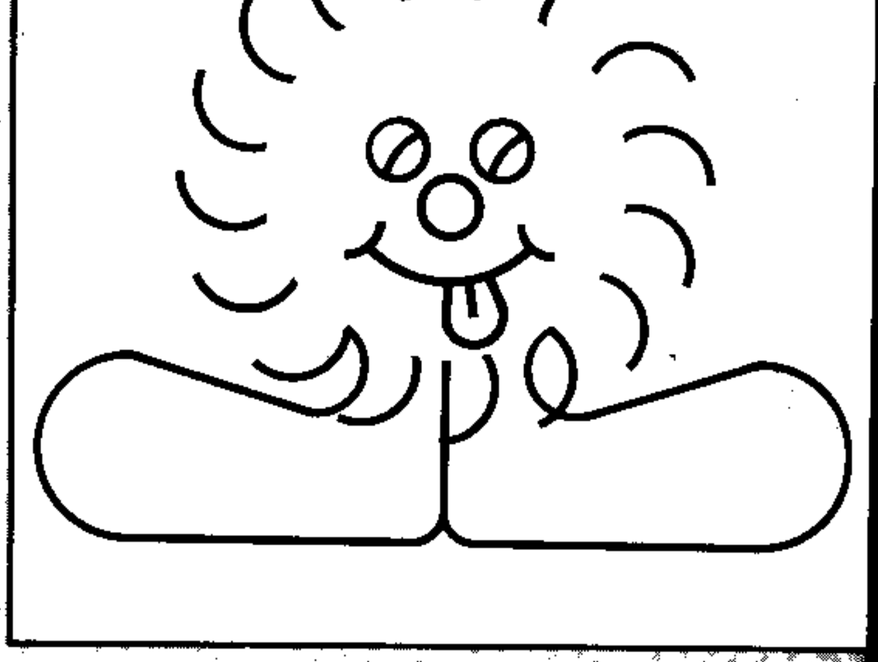
PILOT is available for Atari, Commodore, and Apple computers, and Pascal for Apple, IBM, Texas Instruments, Atari, and Radio Shack computers. Pascal, however, is not cheap. You can pay from $250 to $1,000, depending on the computer model you own.

For information on FORTH, call or write the FORTH Interest Group (P.O. Box 1105, San Carlos, CA 94070; (415) 962-8653). FIG's bimonthly magazine, *Forth Dimensions*, carries lists of FORTH products and vendors; subscription is free with membership in FIG ($15). For information on the Jupiter Ace computer, which has FORTH built in, contact Computer Distribution Assn., 17 S. Main St., Pittsford, NY 14534; (716) 385-6277.

Learning Logo Is a Family Affair

LOGO, THE EDUCATIONAL PROGRAMMING LANGUAGE BASED ON GRAPHICS, WAS DESIGNED FOR YOUNG CHILDREN; BUT IT'S BOTH SIMPLE AND POWERFUL—TRULY A LANGUAGE FOR ALL AGES

BY MINDY PANTIEL AND BECKY PETERSEN



Authors' rendering of Mr. Fuzzy, the creature programmed in Logo by the Weskalnies children.

Meet Mr. Fuzzy. He's a delightful, multi-colored creature that appears on the computer screen when Brian or Sherry Weskalnies enters the command DRAW FUZZY. First a wide circle with green hair appears, then big blue eyes and a red nose—followed by a line for a smile, and, finally, gigantic feet.

Twelve-year-old Sherry and 11-year-old Brian, two pint-sized programmers from Longmont, Colorado, wrote the program for Mr. Fuzzy in a programming language called Logo. Though Logo was designed especially for children and is simple to learn and use, it is also powerful and flexible enough to appeal to computer users of all ages. As members of the Weskalnies family found, Logo offers a little something for everyone.

WHAT'S A PROGRAM? WHAT'S A LANGUAGE?

A computer program is a set of step-by-step instructions that tells the computer how to solve a given problem. Computer programs are written with programming languages, each controlled by its own distinct vocabulary and clearly defined rules. [*See "A Novices' Guide to Programming Languages," on page 54.*]

Logo, or turtle geometry as it's often called, has several alluring features. First, it's simple enough to allow very young children to write their own computer programs; second, because the basis for Logo is graphics, the programs provide immediate visual results; third, despite its apparent simplicity, Logo has the capacity to illustrate complex mathematical ideas.

But the real beauty of Logo lies in its de sign and style. The overall intent of Logo, a stated by Seymour Papert, its chief develope is to allow children to communicate with com puters naturally—more like learning French b living in France, than learning it through tex books.

PARENTS, CHILDREN, AND LOGO

Sherry and Brian, the creators of M Fuzzy, first learned Logo during a six-week summer-school workshop. They worked on a Apple IIe computer with *The Terrapin Log Language* (one of several versions) for 90 mi utes each morning. During the last week classes, they invited the rest of the family join them on parents' night. The main obje tive was to teach their parents everything th had learned. By the end of the evening, bo their mother, Sharon, and three-and-a-ha year-old brother Chris were beginning to u

---

MINDY PANTIEL *and* BECKY PETERSEN, *partners in a communications company, wrote "Take a Lesson from Teachers" for the December issue of* FAMILY COMPUTING.

rstand what Logo was all about.

With coaching from Sherry and Brian, rts learned enough commands to draw a phic on the screen. And Mom moved easily to writing procedures for several different ometric shapes.

Jerald, their father, unable to get to par- s' night, came to class the next morning to d out what he'd missed. Brian and Sherry monstrated some of their fancier programs d then taught him some Logo commands d how they could be used in different ways.

Then Jerald joined in. With the Logo users' nual in hand, the threesome was off and ning, using Logo in ways that the class had t yet tried. They had great fun, and many her class members gathered around to watch e family in action.

NGING IT ALL BACK HOME

The same kind of learning and sharing can on in the home. Children are naturals with go. As with most aspects of computing, chil- n approach programming with a sense of venture and few inhibitions about touching e keyboard or making mistakes in front of hers. Even preschoolers who are just learn- g the alphabet and number symbols can rn Logo's commands with a little parental aching or assistance from slightly older sib- gs.

If elementary school–age children have yed with Logo in class, and are using words ch as *primitives*, *procedures*, and *sprites*, lults shouldn't be intimidated. Children are ally not that far ahead, and—what's even bet- r—they make excellent teachers. With the ta- es turned, newly trained youngsters are usu- y very eager to sit down at the computer and are what they know with an eager adult.

By the same token, teenagers might be e to learn something from their younger others and sisters. Most teenagers who have udied computer programming in school have obably started with a language such as BA- C, which is built into most computers. An troduction to Logo can be just as captivating these teenagers as it is for parents.

Working together at writing and perfecting ograms can provide quality time as a family. is kind of group project can go a long way ward easing whatever tensions between ung and old the computer has caused. And added bonus is that everyone learns to eak the same language.

T THE TURTLE

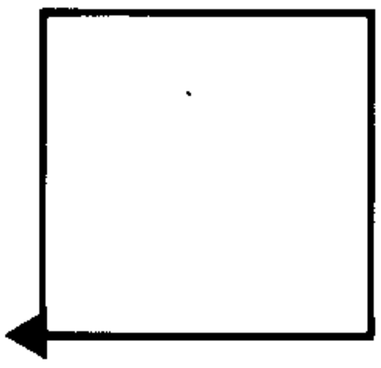
The central figure of Logo is a "turtle," ich rests in the center of the computer reen until told to move elsewhere. The turtle highly mobile and can be directed to take rtle steps" (approximately one eighth of an inch) with such simple commands as FORWARD (FD), BACKWARD (BK), RIGHT (RT), and LEFT (LT). The turtle leaves tracks—which make a line—as it moves.

Using these commands and others, procedures can be written to draw shapes, which can be saved and called back again. When the original shape is back on the screen, you can add to it and develop an increasingly complex program and graphic.

GET STARTED—WITH A SQUARE

When starting out with Logo, one of the first things a programmer might do is make a square. Young children find it helpful to first "play turtle," by walking through a square themselves, before moving to the computer to teach it the same movements. Older children and adults may want to run through their thinking with a pencil and paper first. Through trial and error, family members will quickly learn that a square consists of a series of forward movements and turns. One procedure for a square might look like this:

```
TO SQUARE
   FD 50
   RT 90
   FD 50
   RT 90
   FD 50
   RT 90
   FD 50
END
```



In the procedure TO SQUARE, the programmer is telling the turtle to move forward 50 turtle steps and make a turn to the right of 90 degrees. This is repeated until all four sides of the square are formed. Once the square has been defined, it can be saved and used in combination with other procedures to make pictures of windows, pinwheels, or houses; or changed in size by adding variable lengths; or enhanced by the addition of background or pen colors.

FOUR SQUARES MAKE A WINDOW

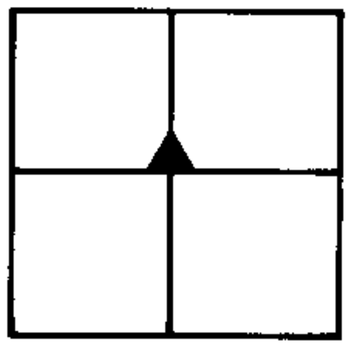
The procedure TO WINDOW combines SQUARE and repeats it four times to make a window.

Shortcuts can be used, too. A command of REPEAT tells the computer to repeat a series of commands written within brackets a specified number of times. The procedure TO WINDOW2 (next page) will execute the exact same graphic as TO WINDOW1.

FAMILY MEMBERS CAN DEVELOP THE SKILLS TO WRITE PROGRAMS FOR GRAPHICS SUCH AS ROCKETS, WINDMILLS, AND FLOWER GARDENS.

The examples of Logo programs shown in this article were written with The Terrapin Logo Language *for the Apple IIe.*

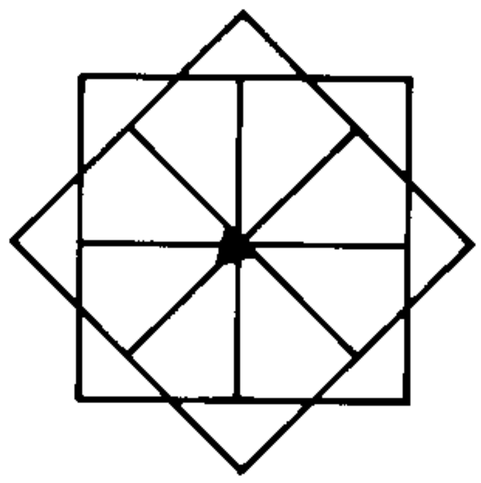
```
TO WINDOW1
   SQUARE
   SQUARE
   SQUARE
   SQUARE
END

TO WINDOW2
   REPEAT 4 [SQUARE]
END
```



If you take the square and turn it eight times, at a 45-degree angle each time, you will create a pinwheel. You can see that with a very limited number of commands you can develop increasingly complex shapes.

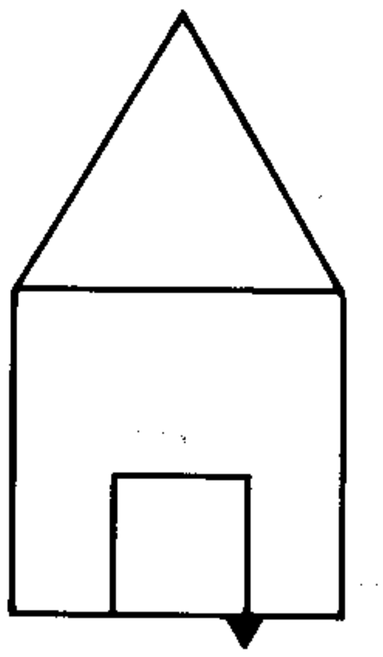
ONCE THE SQUARE HAS BEEN DESIGNED, IT CAN BE USED WITH OTHER PROCEDURES TO DRAW MORE COMPLEX SHAPES.

```
TO PINWHEEL
   REPEAT 8[SQUARE RT 45]
END
```



Three separate procedures are used to draw the house below. SQUARE is used for the main structure. TRI, an equilateral triangle, is used to form the roof, and DOOR is used to form the door. Between each subprocedure, the turtle must be directed to the proper screen location to execute the next part of the graphic. For instance, in the TO HOUSE procedure, after the square is drawn, the program tells the turtle to turn right (RT 90) and go forward (FD 50). The turtle is then in position to draw the triangle, or roof.

```
TO TRI
   RT 30
   FD 50
   RT 120
   FD 50
   RT 120
   FD 50
END

TO DOOR
   FD 20
   RT 90
   FD 20
   RT 90
   FD 20
END

TO HOUSE
   SQUARE
   RT 90
   FD 50
   TRI
   LT 90
   FD 50
   LT 90
   FD 15
   LT 90
   DOOR
   SETH 0
END
```



Though the commands in the programs can be immediately checked on the screen, users soon discover that it's hard to write a program that works on the first try. Consequently, programmers must do a lot of work "debugging"—or fixing—the program. While most people show little tolerance for errors at first, they gradually develop a willingness and even eagerness to find the problems and refine their programs. After all, writing a program is really an exercise in problem solving.

Eventually, family members will develop the skills needed to write computer programs for such complex graphics as rockets, windmills, or flower gardens—complete with color and even animation. Thus, though Logo may seem simple, it always presents new challenges. For instance, when writing the program for Mr. Fuzzy, Sherry and Brian Weskalnies had to deal with geometry, algebra, and general problem solving, and use the fundamentals of structured and modular programming.

IS LOGO FOR EVERYONE?

People who have taught both BASIC and Logo have found that users can learn enough about Logo in the first hour or two to program it right away. Such teachers say that beginning Logo programmers are caught up in the excitement generated by the graphics language, and their anxiety about computers seems to vanish. Teachers say they rarely find this reaction to programming from students first learning BASIC.

Nonetheless, family members should examine why they want to learn to program and what they want to use it for. If the motivation is simply to gain an understanding of what computer programming is like or to discover more about the computer education youngsters may be getting at school, then Logo is a good language to consider.

However, if beginning programmers eventually want to move into professional programming, write programs that can solve real-world problems, or modify commercial software to suit their own needs, then a language other than Logo might be more appropriate.

But such people—especially parents with young children—should still consider learning Logo as a family affair. The programming fundamentals that are mastered through Logo can be put to use in learning the new vocabulary and rules of another computer language. Besides, doesn't everyone have some kind of Mr. Fuzzy floating around in their imagination?

Different Versions of Logo

THEY ALL HAVE TURTLE GRAPHICS, BUT SOME OFFER A LOT MORE

BY KENNETH P. GOLDBERG

At one time, not so long ago, you needed a Texas Instruments 99/4 computer to use the Logo language. It was first developed for that computer. Shortly thereafter, Logo became available for the Apple II plus. Today, the situation has changed drastically. There are five versions of Logo available for Apple computers, and four for the IBM PC. Logo is also available for Atari, Commodore, and Radio Shack computers; and versions are being developed for Timex and Coleco's ADAM.

If you're thinking of buying a microcomputer, and think that Logo may be a language for you or your children, then it's worth considering the various Logo packages when comparing computers. And, if you already own a computer that has several Logo versions available, you should be aware of the differences, so you can choose one that best suits your needs and pocketbook.

The major feature of most Logo packages is the ability to program and see graphic designs on your screen. All versions of Logo provide the user with a screen "turtle" that can be moved about to draw pictures and create designs in a variety of colors. But the various Logos can vary markedly in other respects.

Let's look at some of the characteristics and capabilities that can make one version of Logo differ from another—both in its operation and its possible application. The features of each version are noted in the accompanying chart.

Format (Disk or Cartridge): Most current versions of Logo come in disk form, and consequently require a disk drive. A disk drive also allows you to save procedures and/or graphics, something most users like to do.

The major advantages of a cartridge version are that you don't need a disk drive to use it (and disk drives can cost upwards of $300) and it is less likely that a cartridge will be damaged.

Backup Copy: Many disk versions of Logo come with a backup copy (if the original disk is protected against copying). If not, instructions in your users' manual will explain how to make a backup copy yourself. If you cannot get a backup copy, think seriously about whether you want to risk the full purchase price, for software is not cheap and can be easily damaged.

Cartridge versions of Logo usually don't come with a backup copy. And, generally speaking, you cannot copy a cartridge.

FILL Primitive: If you draw an outline of a shape, and want to fill it in with a color, the FILL command allows you to do that with a single command. ("Primitive" means that the command is built into the program.) You can accomplish the same effect without the FILL command, but it requires writing a procedure and is significantly more time-consuming.

While the FILL command makes it much easier to draw colorful pictures and designs, many educators have reservations about its use, and some developers have chosen not to include it. These critics contend that the FILL command makes the creative process too easy, and that children don't learn as much about programming as they do writing their own coloring procedures. Nonetheless, the FILL capability is a nice feature.

Sound Generation: Although Logo is best known for its graphics capability, it can be used for many nongraphic applications as well. One of these is music composition. Several versions of Logo have a built-in sound-generation

LOGO CAN BE USED FOR MUSIC COMPOSITION, WRITING POETRY, WORD PROCESSING, LIST PROCESSING, AND DOING MATH HOMEWORK.

KENNETH P. GOLDBERG *is also the author of "A Novices' Guide to Programming Languages" in this issue.*

LOGO VERSIONS WITH SPRITES ALLOW YOU TO ANIMATE YOUR PICTURES.

system—which allows the user to modify the pitch, frequency, and duration of sound. With this feature, you can create and play musical compositions. Sound generation can also be used with text and graphics to create exciting games and activities.

Arithmetic: If you want to make learning and using Logo a "family affair," you might want a version that your children can use to do their math homework, and that you can use for simple home and business math applications. Many versions of Logo offer this capability. For instance, to add 15 and 32, all you have to do is type PRINT 15 + 32 and press the RETURN (or ENTER) key, and the answer (47) appears on the screen.

Math Functions: Most Logo versions also include more sophisticated math functions and allow the user to work with trigonometry and logarithms. If you want your children to learn a language that can be used for more than drawing pictures, and possibly as a stepping stone toward learning a more advanced programming language, then you should consider these advanced math functions when comparing programs.

List Processing: List-processing capability means a program can generate and manipulate data and text, such as numbers, letters, words, and sentences. List processing can be used to write poetry or word-and-sentence games; and, it can double as a simple word-processing program to help improve writing and editing skills. List processing can also be combined with Logo's graphics and mathematics capabilities to develop sophisticated games and educational programs.

Sprites: Sprites are screen objects that can be given shape and color and set in motion to animate a picture. They come either as predefined shapes or with a shape-editing procedure that can be used to define new shapes of your own choosing. For example, if CLOUD is a predefined shape, then the command SETSHAPE CLOUD will put a cloud on your screen. In versions that allow you to design your own shapes, you could define a rocketship, and then use the command SETSHAPE ROCKETSHIP to set it in motion from the bottom to the top of the screen, as if it were blasting off. The main thing to know about sprites is how many are available. The more, the better.

As with the FILL command, some educators argue that sprites make it too easy to create a

A QUICK COMPARISON OF

Name	Developer	Machine requirements	Price	Format	Backup copy?	FILL primitive?	S
Apple Logo	Logo Computer Systems Inc. (514) 631-7081	Apple II/II plus/IIe with 64K	$175	Disk	Included	No	
Sprite Logo		Apple II/II plus/IIe	$299	Disk	Permission to copy given	Yes	
Atari Logo		All Atari home computer models	$100	Cartridge	No	No	
IBM Logo		IBM PC/IBM PC*jr*	$175	Disk	Permission to copy given	Yes	
Color Logo	Micropi Inc. available through Radio Shack (817) 390-3944	32K TRS-80 Color Computer with disk drive (disk); 16K TRS-80 Color Computer (cartridge)	Disk $99 ROM pak $49.95	Disk or ROM pak	Permission to copy given	No	
Commodore Logo	Terrapin Inc. (617) 492-8816	Commodore 64	$59.95	Disk	Must send $5 to Commodore	No	
DR Logo	Digital Research Inc. (617) 751-5139	IBM PC (with CP/M 86)	$99.95	Disk	Included	No	Y
Krell Logo (without sprites)	MIT Logo Group (617) 253-7357	Apple II/II plus/IIe; Franklin Ace 1000	$89.95	Disk	Included	No	Y
Krell Logo (with sprites)		Apple II/II plus/IIe	$400	Disk	Included	No	Y
PC Logo	Gold Hill Computer and Harvard Assoc. (617) 492-0660	IBM PC; Eagle PC; Compaq	$199.95	Disk	Included	No	Y
Terrapin Logo	MIT Logo Group (617) 253-7357	Apple II/II plus/IIe with 64K; Franklin Ace	$149.95	Disk	Not included; copy protected	No	Y
TI Logo II	MIT Logo Group and Texas Instr. (617) 253-7357	TI-99/4A with 48K	$99.95	Plug-in module	No	No	Y

colorful, active screen display, and that children won't explore and develop new ideas and programs. But Seymour Papert, the inventor of Logo, thinks that there should be no rules governing the use of the language, and that the different versions will naturally appeal to different types of people.

Setspeed Primitive: The SETSPEED command will set a sprite in motion at a certain speed. This is desirable because you may want different sprite objects to move at different speeds in the same picture. For example, the rocketship blasting off into the clouds would certainly be expected to move faster than the cloud, which should be slowly drifting across the sky. Finally, a SETHEADING command (not included in the chart), which some versions include, will allow you to tell a sprite to move in a certain direction.

Print Out Text and Graphics: There are times—either for record keeping or in order to share your programming exploits with friends—that you may want to print out copies of your work on paper. You may want copies of procedures or programs, data or text material, or graphics. In order to do this, Logo must have the capability built into it. Of course, to print out screen graphics on paper you also need a printer with graphics and/or color printing capability, and not all printers have this. But the first requirement is that the language itself have the ability to print out graphics.

Save and Read Graphics from Disk: All versions of Logo allow you to save text, procedures, and programs on a disk, so that you can read them back into the computer whenever you want to see or use them again. Some versions, however, also allow you to save *graphics* that you have drawn, and then "read" them back into the computer's memory and onto the screen, colors and all. This is not a necessary feature, but it does allow you to call back graphics onto the screen very quickly. You could do it otherwise by recalling the program itself and then running it, which would take a little more time.

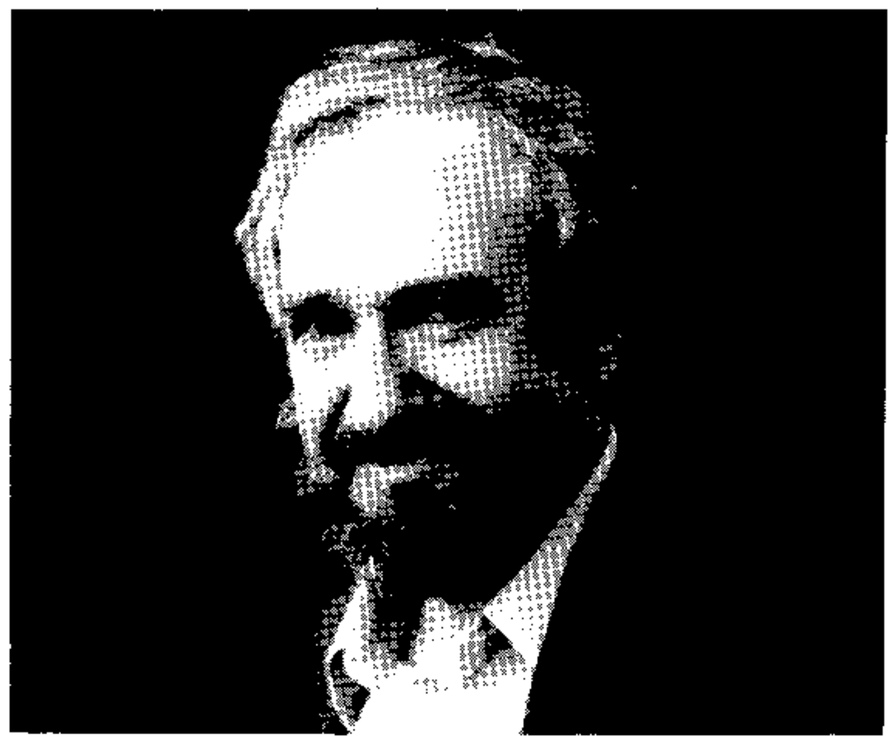
If you want to use Logo to develop activities that mix text and graphics smoothly and without delay (such as stories in which pictures appear on the screen at appropriate times to break the monotony of pure text), or if you want to avoid undue delay in getting your drawing back on the screen from a disk, this is a feature well worth having.

YOU CANNOT PRINT OUT SCREEN GRAPHICS ONTO PAPER UNLESS LOGO HAS THAT CAPABILITY BUILT IN.

OF VERSIONS OF LOGO

Sound?	Arithmetic?	Math functions?	List processing?	Sprites? (how many?)	SETSPEED primitive?	Print out text and graphics?	Save and read text and graphics on disk?
No	Yes	Yes	Yes	No	No	Text only (graphics w/tool kit)	Text only
No	Yes	Yes	Yes	30	Yes	Both	Both
Yes	Yes	Yes	Yes	4	Yes	Text only (program for graphics planned)	Text only
Yes	Yes	Yes	Yes	No	No	Both	Both
No	Yes	No	No	256	Yes, called slow	Text only	Both
Yes	Yes	Yes	Yes	8	Yes	Both	Both
Yes	Yes	Yes	Yes	No	No	Both	Text only
Yes	Yes	Yes	Yes	No	No	Both	Both
Yes	Yes	Yes	Yes	32	No	Both	Both
Yes	Yes	Yes	Yes	No	No	Text: graphics w DOS 2.0	Both
Yes	Yes	Yes	Yes	No	No	Both	Both
Yes	Yes	No	Yes	32	No	Text only	Text only

The Man Behind Logo



BY NICK SULLIVAN

SEYMOUR PAPERT, THE INVENTOR OF LOGO, HAS A LOT OF POWERFUL IDEAS.

"I THINK IT'S BETTER TO PUT 100 COMPUTERS IN ONE PERCENT OF THE SCHOOLS THAN ONE COMPUTER IN 100 PERCENT OF THE SCHOOLS."

Seymour Papert, professor of mathematics and education at the Massachusetts Institute of Technology (MIT), and self-confessed "hacker," is recognized as the founding father of Logo. [See "Making Logo a Family Affair," on page 64]. *He designed the new programming language with the idea that it should be at once powerful and accessible to children and other new computer users. He then formed the MIT Logo Group to develop it. The Logo Group has worked for more than a decade with hundreds of children to develop and refine Logo at MIT's Laboratory for Computer Science. Papert has also worked with Logo Computer Systems Inc. (Quebec, Canada) to produce commercial versions of Logo.*

In recent years, the South African–born Papert has assumed a larger role in the computer world. He has worked with the World Center for Computation and the Human Resources in Paris, which is attempting to cultivate a widespread "computer culture" especially in Third World societies and among unemployed French workers. He is the author of Mindstorms, *a book about "children, computers, and powerful ideas." And he lectures frequently to professional groups, urging them to build and disperse a computer-learning culture in the United States.*

Papert thinks that computers can be flexible tools that put the power of learning and sense of discovery into everyone's hands.

When asked a question, Papert's head shrinks back into his shoulders, his forehead furrows, his eyes close to within a millimeter of shut. In this motionless, pensive pose, he looks like a Rodin sculpture. He looks like a thinker.

While he's thinking, he may muse aloud, "Well, yes, what can you say about that?" That gets his engines going, and he starts thinking aloud—old thoughts, current thoughts, totally original thoughts. Then he stops himself with, "I'm just rambling."

Hidden amongst the ramblings are kernels of wisdom that lead to more questions—more than were asked and probably more than can be answered.

FC: You've said that computers are like pencils, in that they're something every child should have. Why do you think that?

Papert: Computers are more than pencils; they're scratchpads. The pencil is not used only for writing but for biting, doodling, writing illicit notes, and scribbling. Furthermore, pencils are not a toy—they're an adult thing. A child who takes a pencil is appropriating an adult thing, which gives him or her a sense of power.

And in schools, if you want to teach children to write, you give them each a pencil. You don't say, "Here's a pencil, pass it around when you're finished." The same should be true of computers. I think it's better to put 100 computers in one percent of the schools, than one computer in 100 percent of the schools.

FC: Most schools aren't going to buy one computer for each student. What do you consider an acceptable ratio?

Papert: I think that one computer per seven students is a good ratio. But there's only one good answer to the question, and that is: "A computer for every child by 1986," which is the slogan of a new campaign I'm launching.

FC: Obviously, wealthy schools have more computers than poor schools. But do you see much difference in the way computers are used in these schools?

Papert: In wealthy schools, kids use computers in more creative ways. In poorer schools, computers are used largely for drill-and-practice. I don't want to take away from schools that have to do this kind of remedial teaching, but they should be doing other things as well. In other words, physical access to computers doesn't necessarily mean cultural access.

FC: Is programming being taught or learned in different ways?

Papert: There's a "hard" mastery and a "soft"

PHOTOGRAPHS BY WYNTER SNOW

NICK SULLIVAN, *the features editor of* FAMILY COMPUTING, *wrote "Portrait of an Artist as a Software Rebel" for the January issue.*

mastery. A lot of the best progamming is soft, or unstructured, but theorists have grabbed onto the hard, planned, structured approach as the best one.

There are two types of programmers. Some love the detail and the security of their "microworld." Others feel threatened by the limits of programming languages. I've learned this because there are two types of Logo. One uses very precise line drawings; the other is more dynamic and is almost good enough to design a *Pac-Man*–type game. And each type appeals to a different type of person.

FC: How would you describe these people?

Papert: There are "obsessional" and "hysterical" people, what some call left-brained and right-brained people. These two types of people, and the two types of Logo, raise another equity issue. You don't want to produce software that will appeal to only one personality. And almost all schools are biased toward the obsessional, or structured, character, just as they were once biased toward right-handers. We know now that making people organize their thoughts in an unnatural fashion—such as turning left-handers into right-handers—can cause great harm to their learning. The same syndrome is occurring today with computers. Maybe that's why the best programmers don't learn their tricks in school.

FC: A lot of people worry that computers isolate kids from one another and from society at large. They refer to the recent spate of computer break-ins as a case in point. What do you think?

Papert: Left alone, without intervention, there's a pretty large scale of isolationism among computer users. But the popular views comparing hackers to drug addicts are extremely oversimplified. There may be a few hackers who are criminal, just as there are painters or merchants who are criminal. But children and computers . . . it's just a social response to growing up. And the majority of children respond to computers in a social way.

FC: If parents are worried that computers are isolating their children, what can they do about it?

Papert: I don't think there's much they can do. The FBI seizing computers from kids who break into computer systems, as they did last fall, is about as effective as shutting down stills during Prohibition. The only way to stop computer abuse—which is what some see as the result of computer isolation—is to turn the U.S. into a high-security prison.

What I think we'll see is a "privitization" of education. As parents learn that they can do more with computers than the schools are doing, there'll be a tendency to set up alternative social settings, such as block learning groups. They might meet three times a week, or whatever, as computer users' groups do.

FC: Do you see much difference in the way boys and girls respond to computers?

Papert: I don't want to make too many sweeping generalizations. Let's just say that girls adapt to Logo more easily than they do to BASIC; and the difference between girls and boys is much smaller with Logo.

FC: Do you think your voice, or anyone else's, is being heard—are computers taking hold among the underprivileged?

Papert: There's nothing systematic so far. There's a lot of grass-roots activity with drop-in centers and the like. What's most exciting about the computer movement is the number of people doing new things—writing software, starting magazines, writing letters. It's unprecedented, effervescent learning. But, to make this learning widespread, we need the help of computer professionals. I think that in time we may see some of the unrest typical of the 1960s, with people symbolically banging down the doors of decision makers to protest against discrimination in society.

FC: Where do you stand on the issue of piracy—or illegally copying software and distributing it to others?

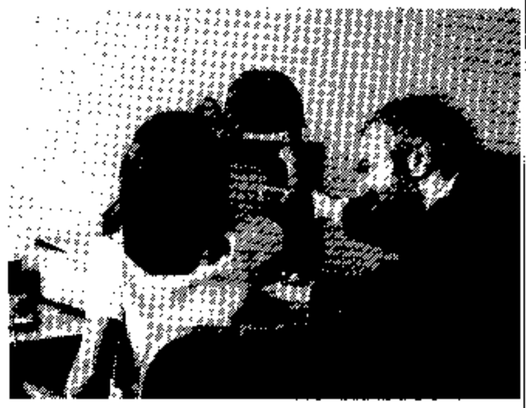
Papert: I'm schizophrenic. I believe, in principle, that software should be distributed as a free resource; on the other hand, it costs money to develop software. I have a huge conflict with Logo. I'd like to give it away. But unless a gold mine is found, we'd have no money to support new development.

FC: Did you ever think, 15 years ago, that Logo would be so successful?

Papert: No, the whole thing took me by surprise. The most surprising part has been the way it's taking hold. I had thought Logo would succeed, but very slowly, by trickling down from academics and theorists to the schools. But it's been turned upside down. The teachers and students have been pushing for it.

FC: In *Mindstorms*, you say that your fascination with gears as a child, and your eventual understanding of how they worked, gave you a model to grasp new and abstract ideas. Do you think Logo has succeeded in giving other children a flexible model of their own?

Papert: Yes, I think so, for quite a few children. It's quite pleasing to see children five years later who have integrated their Logo experiences into their thought patterns. That's very gratifying to me. FC



Dr. Seymour Papert introducing a youngster to the fun of Logo.

"I HAVE A HUGE CONFLICT WITH LOGO. I'D LIKE TO GIVE IT AWAY. BUT UNLESS A GOLD MINE IS FOUND, WE'D HAVE NO MONEY TO SUPPORT NEW DEVELOPMENT."

A SAMPLING OF DATA-BASE SYSTEMS

Choosing a data-base management system requires careful thought and research. Although the initial time investment may seem excessive, it's better than finding you've wasted your cash investment later. Your first step before purchasing a data-base program is to determine its potential applications for you. Ask yourself some basic questions: Will children be using it? How often will you be using it? Review the program's features: Can it search, sort, or formulate? Visit local retailers and have them demonstrate their software with your applications in mind. Remember: It is more effective to choose a system based on your current needs than to buy one and try to adapt it. We have provided a brief list of software available for use on different machines. [*An in-depth feature about data-base- management programs and how to shop for them will appear in an upcoming issue.*]

PROGRAM	COMPANY	MACHINES	PRICE
dBaseII	ASHTON-TATE, 10150 W. Jefferson Blvd., Culver City, CA 90230, (213) 204-5570	Apple II/II plus/IIe/III w/CPM, IBM PC, TRS-80 Model II,	$700
Data Base Management	ALPHA SOFTWARE CORP., 30 B St., Burlington, MA 01803, (617) 229-2924	IBM PC	$250
Data Factory	MICRO LAB, 2699 Skokie Valley Rd., Highland Park, IL 60035, (312) 433-7550	Apple II/II plus/IIe IBM PC	$180 $250
Data Perfect	LJK ENTERPRISES, 7852 Big Bend Blvd., St. Louis, MO 63119, (314) 962-1855	Apple II/II plus/IIe Atari 400/800/1200	$129 $99
Database Manager	MICRO ARCHITECT, INC., 6 Great Pine Ave., Burlington, MA 01803, (617) 273-5658	IBM PC TRS-80 Model I/II/16	$195
Database Manager	SIMPLIFIED SOFTWARE SYS., 118 Third Ave., NW, P.O. Box 1192, Hickory, NC 28601, (704) 328-2386	TRS-80 Models I/III	$69
Easy Filer	IUS, 2401 Marinship Way, Sausalito, CA 94965, (415) 331-6700	TI-99/4A IBM PC	$400
File Clerk	SOFTWARE LABS, INC., 6924 Riverside Dr., Dublin, OH 43017, (614) 889-5083	IBM PC	$50
Flexfile	AB COMPUTERS, 252 Bethlehem Pike, Colmar, PA 18915, (215) 822-7727	Commodore 64/VIC-20	$110
General Manager	SIERRA ON-LINE, Sierra On-Line Bldg., Coarsegold, CA 93614, (209) 683-6858	Apple II/II plus/IIe	$229
IBM 5	MICRO ARCHITECT, INC., 6 Great Pine Ave., Burlington, MA 01803, (617) 273-5658	IBM PC TRS-80 Models I/III/4	$99
Mini Jini Record Keeper	JINI MICRO-SYSTEMS, Box 274, Kingsbridge Station, Riverdale, NY 10463, (212) 796-6200	Commodore 64/VIC-20	$89
The Organizer	TIMEX COMPUTER CORP., P.O. Box 2655, Waterbury, CT 06725, (203) 573-5000	TS 1000/1500	$16
PFS: File	SOFTWARE PUBLISHING, 1901 Landings Dr., Mountain View, CA 94043, (415) 962-8910	IBM PC, TI-99/4A Apple II/II plus/IIe Apple III	$140 $125 $175
Profile III plus	TANDY CORP., 400 Atrium, One Tandy Center, Ft. Worth, TX 76102, (817) 338-2395	TRS-80 Models III/4 in III mode	$199
VisiDex	VISICORP, 2895 Zanker Rd., San Jose, CA 95134, (408) 946-9000	Apple II/II plus/IIe TI-99/4A	$250
VisiFile	VISICORP, 2895 Zanker Rd., San Jose, CA 95134, (408) 946-9000	Apple II/II plus/IIe, IBM PC, TI-99/4A	$250

TRIBUTIONS field, specifying all donations greater than $25. You can also search on multiple fields. For instance, I needed lists of everyone who contributed more than $25 and resided outside of the district, and everyone who endorsed the candidate and gave money as well.

"When the Girl Scouts had their cookie sale, I was the neighborhood cookie manager. I created a data base that had a record for each troop's cookie managers, including name, address, and phone number. I specified a field for each kind of cookie. All I had to do was enter how many cases of cookies a troop ordered, and then multiply the number of cases times the cost of the case and make a new field out of the total. There were 22 troops. A one-page report listed every troop number and how much profit was shown.

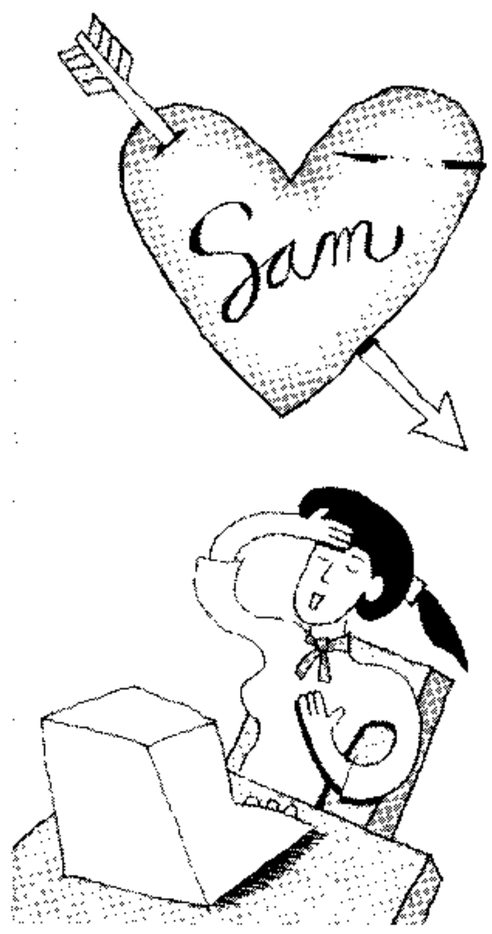
"The data base takes a while to set up, and then you just have to update it. All it takes is time—time and typing," Karen corrected herself. "The work comes in cycles—the beginning of the school year, election times: then it goes away. For periods of weeks, I may not touch the data bases. But if you don't update regularly, you might as well throw it in the trash."

Data-base management can become a catchy activity; it did in Karen's neighborhood. "As a result of seeing what our computer does, Susan's Girl Scout leader has bought an Apple," says Karen. "I can step back and get involved in something else . . ." Karen said, en route to showing the troop leader how to hook up her new printer.

PERSONAL VALENTINE

BY JOEY LATIMER

It's Valentine's Day. Cancel the order for a dozen roses, return the chocolate hearts, and throw away the sappy cards. Turn your marvel of technology into a computing Cupid. Design a personalized message for that special someone.



ADAM/*Personal Valentine*

```
100 HOME
120 PRINT "COMPUTER VALENTINE"
130 PRINT
140 PRINT "PRESS <RETURN> AFTER EACH REPLY."
160 PRINT
170 INPUT "WHAT IS YOUR NAME? ";n$
180 PRINT
190 PRINT "WHO IS THIS VALENTINE FOR?"
200 INPUT "(8 letters or fewer, please)";f$
210 IF LEN(f$)>8 THEN 200
240 GR
250 COLOR=6
260 FOR i=0 TO 39
270 VLIN 0,39 AT i
280 NEXT i
290 COLOR=9
300 l=3
310 GOSUB 2000
320 COLOR=4
330 l=11
340 GOSUB 2000
350 COLOR=0
360 HLIN 12,29 AT 37
370 HLIN 12,29 AT 36
380 HLIN 3,8 AT 13
390 HLIN 33,38 AT 13
400 HLIN 18,23 AT 5
410 COLOR=2
420 VLIN 36,37 AT 19
430 VLIN 36,37 AT 21
440 VTAB 20
450 HTAB 5
630 PRINT "MY HEART THROBS FOR YOU"
640 HTAB 15
650 PRINT f$;"!"
660 HTAB 19-(LEN(n$)/2)
670 PRINT "Love, ";n$
680 COLOR=INT(RND(1)*16)
700 GOSUB 1000
710 COLOR=9
720 GOSUB 1000
730 GOTO 680
1000 HLIN 17,18 AT 10
1010 HLIN 23,24 AT 10
1020 HLIN 17,18 AT 11
1030 HLIN 23,24 AT 11
1040 HLIN 16,19 AT 12
1050 HLIN 22,25 AT 12
1060 HLIN 16,19 AT 13
1070 HLIN 22,25 AT 13
1080 HLIN 16,25 AT 14
1090 HLIN 16,25 AT 15
1100 HLIN 17,24 AT 16
1110 HLIN 17,24 AT 17
1120 HLIN 18,23 AT 18
1130 HLIN 18,23 AT 19
1140 HLIN 18,23 AT 20
1150 HLIN 18,23 AT 21
1160 HLIN 19,22 AT 22
1170 HLIN 19,22 AT 23
1180 HLIN 20,21 AT 24
1190 HLIN 20,21 AT 25
1200 RETURN
2000 FOR i=1 TO l
2010 READ x1,y1,x2,y2
2020 FOR j=x1 to x2
2030 VLIN y1,y2 at j
2040 NEXT j
2050 NEXT i
2060 RETURN
3000 DATA 18,0,22,5,4,14,6,24,34,14,36,24,12,5,28,39,3
3010 DATA 11,7,13,4,9,10,11,6,7,8,9,8,6,10,9,10,5,12
3020 DATA 10,33,11,37,13,34,9,36,11,32,7,34,11,30,6,32
3030 DATA 11,28,5,30,10
```

Apple/*Personal Valentine*

```
100 HOME
120 PRINT "COMPUTER VALENTINE"
130 PRINT
140 PRINT "PRESS <RETURN> AFTER EACH REPLY."
160 PRINT
170 INPUT "WHAT IS YOUR NAME? ";N$
180 PRINT
190 PRINT "WHO IS THIS VALENTINE FOR?"
200 INPUT "(8 LETTERS OR FEWER, PLEASE) ";F$
210 IF LEN(F$)>8 THEN 200
240 GR
250 COLOR=6
260 FOR I=0 TO 39
270 VLIN 0,39 AT I
280 NEXT
290 COLOR=9
300 L=3
310 GOSUB 2000
320 COLOR=4
330 L=11
340 GOSUB 2000
350 COLOR=0
360 HLIN 12,28 AT 37
370 HLIN 12,28 AT 36
380 HLIN 2,6 AT 14
390 HLIN 34,38 AT 14
400 HLIN 18,22 AT 5
410 COLOR=2
420 VLIN 36,37 AT 19
430 VLIN 36,37 AT 21
440 VTAB 23
450 FLASH
460 HTAB 8 - (LEN(F$)/2)
630 PRINT "MY HEART THROBS FOR YOU, ";F$;"!"
640 PRINT
650 HTAB 19 - (LEN(N$)/2):PRINT "LOVE, ";N$
680 COLOR=INT(RND(1)*16)
700 GOSUB 1000
710 COLOR=9
720 GOSUB 1000
730 GOTO 680
1000 HLIN 17,18 AT 10
1010 HLIN 23,24 AT 10
```

```
540 FOR P=1 TO 125
550 NEXT P
560 IF INKEY$ = CHR$(27) THEN END
570 GOTO 520
1000 FOR I = 1 TO L
1010 READ X1,Y1,X2,Y2
1020 LINE (X1,Y1)-(X2,Y2),C,BF
1030 NEXT I
1040 RETURN
2000 DATA 56,91,80,114,220,91,244,114,110,45,190,150
2010 DATA 110,56,80,1.57,3.14,80,85,190,220,244,6.28
2020 DATA 1.57,220,85,140,44,160,41,56,83,81,86,219
2030 DATA 83,244,86,110,136,190,140
```

TI-99/4A/*Personal Valentine*

```
10 A$="8142241818244281"
20 B$="FFFFFFFFFFFFFFFF"
30 CALL CHAR(128,B$)
40 CALL CHAR(136,B$)
50 CALL CHAR(144,A$)
60 CALL COLOR(12,13,16)
70 CALL COLOR(13,11,11)
80 CALL COLOR(14,7,7)
90 CALL COLOR(15,3,8)
100 CALL CLEAR
120 PRINT "COMPUTER VALENTINE"
130 PRINT
140 PRINT "PRESS <ENTER> AFTER"
150 PRINT "EACH REPLY."
160 PRINT
170 PRINT "WHAT IS YOUR FIRST NAME?"
180 PRINT "(7 LETTERS OR FEWER, PLEASE)"
190 INPUT N$
200 IF LEN(N$)>7 THEN 180
210 PRINT
220 PRINT "WHO IS THE VALENTINE FOR?"
230 INPUT F$
240 CALL CLEAR
300 READ CH,A,B
310 IF CH=-1 THEN 630
320 FOR CO=A TO B
330 READ RO,REP
340 CALL VCHAR(RO+3,CO,CH,REP)
350 NEXT CO
360 GOTO 300
630 PRINT F$
640 PRINT TAB(10);"MY HEART"
650 PRINT TAB(7);"THROBS FOR YOU"
660 PRINT TAB(22);N$;
670 CALL SCREEN(9)
700 HO=KO
710 KO=INT(RND*13)+2
720 IF KO=HO THEN 710
730 CALL COLOR(14,KO,KO)
740 CALL SOUND(500,-8,0)
750 GOTO 700
2000 DATA 128,4,7,14,7,14,7,14,7,14,7,128,25,28,14,7
2010 DATA 14,7,14,7,14,7,128,13,19,1,2,1,2,1,2,1,2,1,2
2020 DATA 1,2,1,2,144,3,29,8,5,6,7,5,8,4,9,4,9,3,10,3
2030 DATA 8,3,18,3,18,3,18,3,18,3,18,3,18,3,18,3,18,3
2040 DATA 18,3,18,3,18,3,18,3,18,3,8,3,10,4,9,4,9,5,8
2050 DATA 6,7,8,5,136,13,21,6,2,5,4,5,5,5,6,6,6,5,6,5
2060 DATA 5,5,4,6,2,120,3,8,13,1,13,1,13,1,13,1,13,1
2070 DATA 13,1,120,24,29,13,1,13,1,13,1,13,1,13,1,13,1
2080 DATA 120,13,19,3,1,3,1,3,1,3,1,3,1,3,1,3,1
2090 DATA -1,-1,-1
```

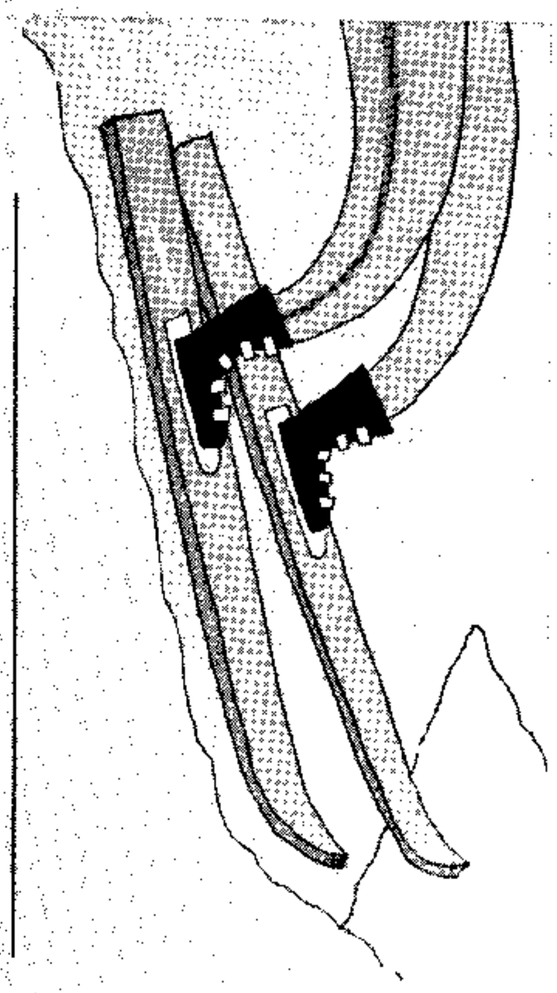
Timex Sinclair 1000 w/16K RAM Pack & Timex Sinclair 1500/*Personal Valentine*

```
100 CLS
110 SLOW
120 PRINT "COMPUTER VALENTINE"
130 PRINT
```

→

SKI TREK

BY JOEY LATIMER



Are you appalled by standing in lift lines for hours to ski a three-minute run? Does just the thought of 30°-below-zero weather send shivers up and down your spine? Then pack away your skis, poles, and thermal underwear, pull up your favorite chair, pour a cup of hot chocolate, and bring out your hardware. Get ready for a tricky trek down a slick, tree-covered course. Remember, there's no snowplowing down this one. Ready? Whoooossshhh!

ADAM/*Ski Trek*

```
80 HOME
130 PRINT TAB(9);"THE SKI GAME"
140 FOR t=1 TO 1500
150 NEXT t
170 HOME
180 PRINT "YOU ARE THE SKIER: H"
190 PRINT
200 PRINT "FOR A HIGH SCORE,"
210 PRINT "SKI DOWN THE SLOPE"
220 PRINT "WITHOUT HITTING TREES."
230 PRINT
240 PRINT "USE THE JOYSTICK TO MOVE LEFT"
250 PRINT "OR RIGHT."
270 PRINT
280 PRINT "PRESS (RETURN) TO BEGIN."
290 INPUT r$
300 HOME
330 PRINT "  GET READY!"
340 FOR t=1 TO 250
350 NEXT t
360 l=12
370 s=0
380 p=PDL(3)
400 a=INT(RND(1)*16)
410 FOR x=1 TO a
420 GOSUB 1000
440 PRINT TAB(x);CHR$(33);"                ";CHR$(33)
450 IF l<=x THEN 2000
460 NEXT x
470 FOR y=a TO 1 STEP -1
480 GOSUB 1000
500 PRINT TAB(y);CHR$(33);"                ";CHR$(33)
510 IF l>=y+12 THEN 2000
520 NEXT y
530 GOTO 400
1000 m=PDL(3)
1010 IF m<p THEN l=l-1
1020 IF m>p THEN l=l+1
1030 p=m
1040 HTAB l
1050 PRINT "H"
1060 s=s+1
1070 FOR t=1 TO 10
1080 NEXT t
1100 PRINT CHR$(8);CHR$(32)
1120 RETURN
2000 REM
2010 PRINT TAB(l);"*"
2030 PRINT
2050 PRINT "  C R A S H!"
2070 PRINT
2090 PRINT "YOU WENT ";s;"YARDS."
2100 IF s>h THEN h=s
2120 PRINT "THE HIGH SCORE IS";h;"."
2140 PRINT
2160 PRINT "PRESS (RETURN) TO PLAY AGAIN."
2180 INPUT r$
2210 GOTO 300
```

Apple/*Ski Trek*

```
80 HOME
130 PRINT TAB(12)"THE SKI GAME"
140 FOR T = 1 TO 1500
150 NEXT T
170 HOME
180 PRINT "YOU ARE THE SKIER: H"
190 PRINT
200 PRINT "FOR A HIGH SCORE"
210 PRINT "SKI DOWN THE SLOPE"
220 PRINT "WITHOUT HITTING TREES."
230 PRINT
240 PRINT "PRESSING..."
250 PRINT "B MOVES YOU LEFT;"
260 PRINT "N MOVES YOU RIGHT."
270 PRINT
280 PRINT "PRESS (RETURN) TO BEGIN."
290 INPUT R$
300 HOME
330 PRINT " GET READY!"
340 FOR T = 1 TO 2500
350 NEXT T
360 L = 12
370 S = 0
400 A = INT(RND(1)*24)
410 FOR X = 1 TO A
420 GOSUB 1000
440 PRINT TAB(X); CHR$(33);"                ";CHR$(33)
450 IF L <= X THEN 2000
460 NEXT X
470 FOR Y = A TO 1 STEP -1
480 GOSUB 1000
500 PRINT TAB(Y); CHR$(33);"                ";CHR$(33)
510 IF L >=Y + 12 THEN 2000
520 NEXT Y
530 GOTO 400
1000 M = PEEK(-16384)
1010 IF M = 194 THEN L = L - 1
1020 IF M = 206 THEN L = L + 1
1040 HTAB(L)
1050 PRINT "H"
1060 S = S + 1
1070 FOR T = 1 TO 20
1080 NEXT T
1100 PRINT CHR$(8); CHR$(32)
1120 RETURN
2000 REM
2010 PRINT TAB(L)"*"
2030 PRINT
2050 PRINT "  CRASH!"
2060 PRINT
2070 PRINT
2090 PRINT "YOU WENT "S" YARDS."
2100 IF S > H THEN H = S
2120 PRINT "THE HIGH SCORE IS "H"."
2140 PRINT
2160 PRINT "PRESS (RETURN) TO PLAY AGAIN."
2180 INPUT R$
2210 GOTO 300
```

```
270 PRINT
280 PRINT "PRESS (RETURN) TO BEGIN."
290 INPUT R$
300 PRINT CHR$(147)
310 POKE 36879,25
320 POKE 650,128
330 PRINT "  GET READY!"
340 FOR T=1 TO 2500
350 NEXT T
360 L=10
370 S=0
400 A=INT(RND(1)*9)
410 FOR X=1 TO A
420 GOSUB 1000
440 PRINT TAB(X)CHR$(30);CHR$(94);"                ";CHR$(
94)
450 IF L<=X THEN 2000
460 NEXT X
470 FOR Y=A TO 1 STEP -1
480 GOSUB 1000
500 PRINT TAB(Y)CHR$(30);CHR$(94);"                ";CHR$(
94)
510 IF L>Y+12 THEN 2000
520 NEXT Y
530 GOTO 400
1000 GET M$
1010 IF M$="B" THEN L=L-1
1020 IF M$="N" THEN L=L+1
1050 PRINT TAB(L)CHR$(144)"H";
1060 S=S+1
1070 FOR T=1 TO 40
1080 NEXT T
1100 PRINT CHR$(20)
1120 RETURN
2000 REM
2010 PRINT TAB(L)CHR$(28)"*"
2030 PRINT
2050 PRINT CHR$(156)"  C R A S H!"
2070 PRINT
2090 PRINT "YOU WENT "S" YARDS"
2100 IF S>H THEN H=S
2120 PRINT "HIGH SCORE IS"H"."
2140 PRINT
2160 PRINT "PRESS (RETURN)"
2170 PRINT "TO PLAY AGAIN."
2180 INPUT R$
2190 PRINT CHR$(159)
2200 POKE 53281,6
2210 GOTO 300
```

IBM PC/*Ski Trek*

```
80 RANDOMIZE (0)
90 CLS
100 SCREEN 0,0,0
110 WIDTH 40
120 COLOR 7,0,0
130 KEY OFF
140 PRINT TAB(14);"THE SKI GAME"
150 FOR T = 1 TO 1500
160 NEXT T
170 CLS
180 PRINT "YOU ARE THE SKIER: H"
190 PRINT
200 PRINT "FOR A HIGH SCORE,"
210 PRINT "SKI DOWN THE SLOPE"
220 PRINT "WITHOUT HITTING TREES."
230 PRINT
240 PRINT "PRESSING ..."
250 PRINT "B MOVES YOU LEFT;"
260 PRINT "N MOVES YOU RIGHT."
270 PRINT
280 PRINT "PRESS (ENTER) TO BEGIN."
290 INPUT R$
300 CLS
330 PRINT"  GET READY!"
340 FOR T = 1 TO 2500
350 NEXT T
360 L = 12
370 S = 0
400 A = INT(RND*(30+1))
410 FOR X = 1 TO A
420 GOSUB 1000
440 PRINT TAB(X);CHR$(24);"                ";CHR$(24)
450 IF L <= X THEN 2090
460 NEXT X
470 FOR Y = A TO 1 STEP -1
480 GOSUB 1000
500 PRINT TAB(Y);CHR$(24);"                ";CHR$(24)
510 IF L >= Y + 12 THEN 2090
520 NEXT Y
530 GOTO 400
1000 M$ = INKEY$
1010 IF M$ = "B" THEN L = L - 1
1020 IF M$ = "N" THEN L = L + 1
1050 PRINT TAB(L);"H"
1060 S = S + 1
1070 FOR T = 1 TO 20
1080 NEXT T
1120 RETURN
2000 REM
2010 PRINT TAB(L);"*"
2030 PRINT
2050 PRINT "  CRASH!"
2090 PRINT "YOU WENT ";S;" YARDS."
2100 IF S > H THEN H = S
2120 PRINT "THE HIGH SCORE IS";H;"."
2140 PRINT
2160 PRINT "PRESS (ENTER) TO PLAY AGAIN."
2180 INPUT R$
2210 GOTO 300
```

TI-99/4A w/TI Extended BASIC/*Ski Trek*

```
10 RANDOMIZE
80 CALL CLEAR
90 CALL CHAR(96,"00183C7EFF181818")
100 CALL COLOR(9,4,1)
130 PRINT TAB(8);"THE SKI GAME"
140 FOR T=1 TO 500
150 NEXT T
170 CALL CLEAR
180 PRINT "YOU ARE THE SKIER: H"
190 PRINT
200 PRINT "FOR A HIGH SCORE,"
210 PRINT "SKI DOWN THE SLOPE"
220 PRINT "WITHOUT HITTING TREES."
230 PRINT
240 PRINT "PRESSING..."
250 PRINT "B MOVES YOU LEFT;"
260 PRINT "N MOVES YOU RIGHT."
270 PRINT
280 PRINT "PRESS (ENTER) TO BEGIN."
290 INPUT R$
300 CALL CLEAR
330 PRINT "  GET READY!"
340 FOR T=1 TO 800
350 NEXT T
360 L=7
370 S=0
380 CALL SCREEN(16)
390 PRINT
400 A=INT(RND*13)+1
410 FOR X=1 TO A
420 GOSUB 1000
440 PRINT TAB(X);CHR$(96);"                ";CHR$(96)
450 IF L<=X THEN 2000
460 NEXT X
```

```
470 FOR Y=A TO 1 STEP -1
480 GOSUB 1000
500 PRINT TAB(Y);CHR$(96);"              ";CHR$(96)
510 IF L>=Y+12 THEN 2000
520 NEXT Y
530 GOTO 400
1000 CALL KEY(3,M,ST)
1010 IF M=66 THEN L=L-1
1020 IF M=78 THEN L=L+1
1030 CALL HCHAR(23,L,72,1)
1060 S=S+1
1070 FOR T=1 TO 20
1080 NEXT T
1100 CALL HCHAR(23,L,32,1)
1120 RETURN
2000 REM
2010 PRINT TAB(L);"*"
2030 PRINT
2050 PRINT "  C R A S H!"
2090 PRINT "YOU WENT ";S;" YARDS."
2100 IF S>H THEN H=S
2120 PRINT "HIGH SCORE IS ";H
2160 PRINT "PRESS (ENTER) TO PLAY AGAIN."
2180 INPUT R$
2210 GOTO 300
```

Timex Sinclair 1000 w/16K RAM Pack & Timex Sinclair 1500/*Ski Trek*

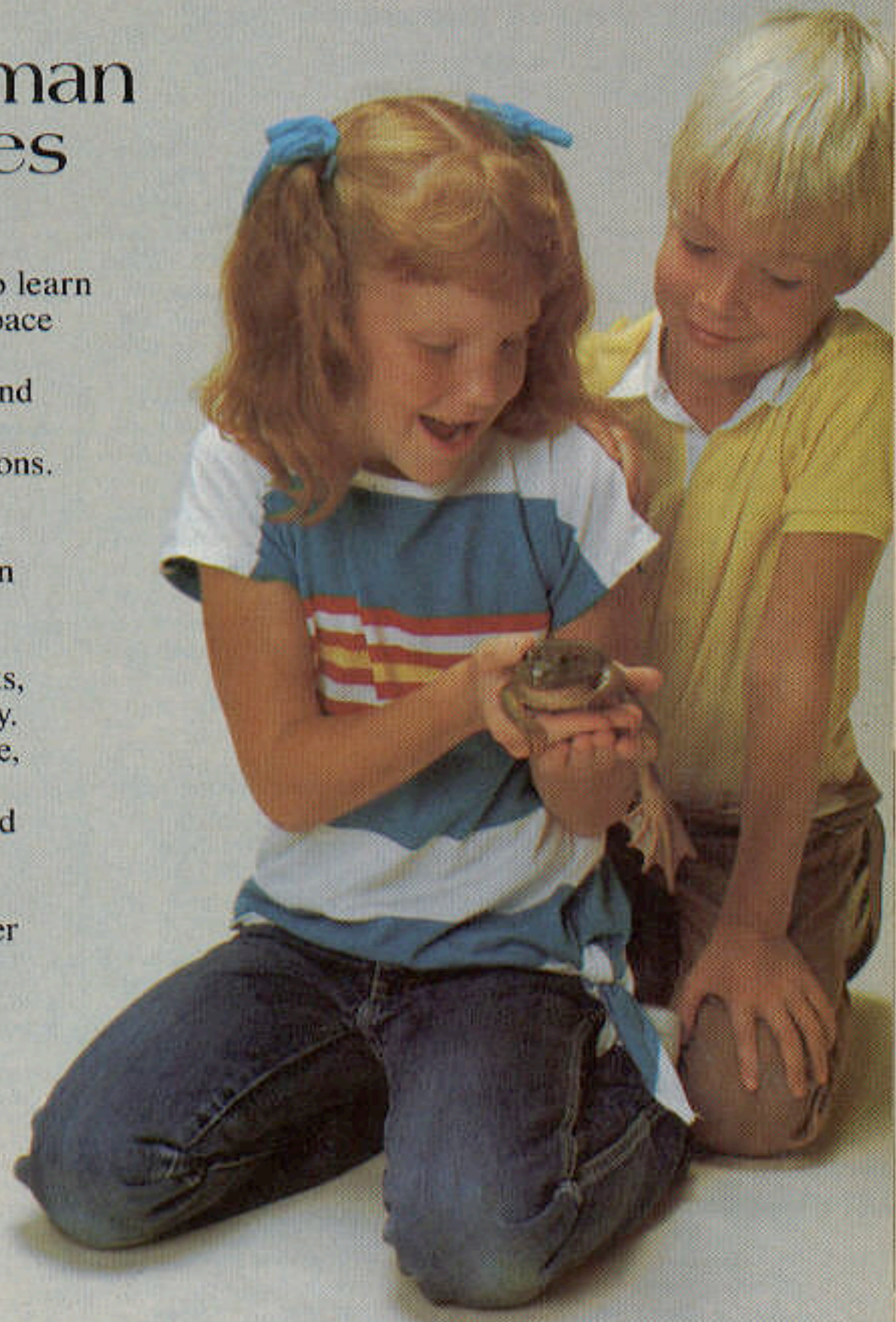
```
10 RAND
70 SLOW
80 CLS
130 PRINT TAB(10):"THE SKI GAME"
140 FOR T= 1 TO 90
150 NEXT T
160 LET H=0
170 CLS
180 PRINT "YOU ARE THE SKIER: H"
190 PRINT
200 PRINT "FOR A HIGH SCORE,"
210 PRINT "SKI DOWN THE SLOPE"
220 PRINT "WITHOUT HITTING TREES."
230 PRINT
240 PRINT "PRESSING ..."
250 PRINT "B MOVES YOU LEFT;"
260 PRINT "N MOVES YOU RIGHT."
270 PRINT
280 PRINT "PRESS (ENTER) TO BEGIN."
290 INPUT R$
300 CLS
330 PRINT " GET READY"
340 FOR T = 1 TO 150
350 NEXT T
360 LET L = 12
370 LET S = 0
400 LET A = INT(RND*18)
410 FOR X = 1 TO A
420 GOSUB 1000
430 IF S > 10 THEN SCROLL
440 PRINT TAB(X);CHR$ 24;"              ";CHR$ 24
450 IF L <= X THEN GOTO 2000
460 NEXT X
470 FOR Y = A TO 1 STEP -1
480 GOSUB 1000
490 IF S > 10 THEN SCROLL
500 PRINT TAB(Y);CHR$ 24;"              ";CHR$ 24
510 IF L >= Y + 12 THEN GOTO 2000
520 NEXT Y
530 GOTO 400
1000 LET M$ = INKEY$
1010 IF M$ = "B" THEN LET L = L - 1
1020 IF M$ = "N" THEN LET L = L + 1
1040 IF S > 10 THEN SCROLL
1050 PRINT TAB(L);"H"
1060 LET S = S + 1
1100 PRINT AT 24-(PEEK 16442),32-(PEEK 16441);" ";
1120 RETURN
2000 SCROLL
2010 PRINT TAB(L);"*"
2020 SCROLL
2030 PRINT
2040 SCROLL
2050 PRINT "  CRASH!"
2060 SCROLL
2070 PRINT
2080 SCROLL
2090 PRINT "YOU WENT ";S;" YARDS."
2100 IF S > H THEN LET H = S
2110 SCROLL
2120 PRINT "THE HIGH SCORE IS ";H;"."
2130 SCROLL
2140 PRINT
2150 SCROLL
2160 PRINT "PRESS (ENTER) TO PLAY AGAIN."
2180 INPUT R$
2210 GOTO 300
```

TRS-80 Color Computer/*Ski Trek*

```
80 CLS
130 PRINT @ 41,"THE SKI GAME"
140 FOR T=1 TO 1500
150 NEXT T
170 CLS
180 PRINT@33,"YOU ARE THE SKIER: H"
190 PRINT
200 PRINT "FOR A HIGH SCORE,"
210 PRINT "SKI DOWN THE SLOPE"
220 PRINT "WITHOUT HITTING TREES."
230 PRINT
240 PRINT "PRESSING..."
250 PRINT "B MOVES YOU LEFT;"
260 PRINT "N MOVES YOU RIGHT."
270 PRINT
280 PRINT "PRESS (ENTER) TO BEGIN."
290 INPUT R$
300 CLS
330 PRINT "  GET READY!"
340 FOR T=1 TO 2500
350 NEXT T
360 L=5
370 S=0
400 A=RND(19)
410 FOR X=1 TO A
420 GOSUB 1000
440 PRINT @ (X+480),CHR$(33);"              ";CHR$(33)
450 IF L<=X THEN 2010
460 NEXT X
470 FOR Y=A TO 1 STEP -1
480 GOSUB 1000
500 PRINT @ (Y+480),CHR$(33);"              ";CHR$(33)
510 IF L>=Y+12 THEN 2010
520 NEXT Y
530 GOTO 400
1000 M$=INKEY$
1010 IF M$="B" THEN L=L-1
1020 IF M$="N" THEN L=L+1
1050 PRINT @ (L+480),"H";
1060 S=S+1
1070 FOR T=1 TO 120
1080 NEXT T
1100 PRINT CHR$(8)
1120 RETURN
2000 REM
2010 PRINT TAB(L);"*"
2030 PRINT
2050 PRINT "  C R A S H!"
2070 PRINT
```

GET THE JUMP ON MATH

With Scott, Foresman Math Action Games



Frog Jump is a great way for your children to learn to work with numbers. Or they can take a Space Journey to learn how to work with percents. Pyramid Puzzler, Star Maze, Picture Parts, and Number Bowling help with multiplication... division... basic facts... decimals and fractions.

Math Action Games get youngsters deeply involved in the excitement of discovering and mastering fundamental mathematics. Children have fun and learn at the same time.

There's a **Math Action Game** for every age group. Exciting formats challenge participants, moving them through three levels of difficulty. Games can be competitive or non-competitive, so children play them again and again. Each game uses color, music, animation, and sound effects to trigger quick thinking and accurate response.

Math Action Games can give your youngster the incentives and the satisfaction he needs to master math. It's an ideal supplement to any school mathematics program. Games are available for most popular microcomputers.

SCOTT, FORESMAN...
Products with tomorrow in mind.

For fast information about MATH ACTION GAMES, call TOLL FREE: 800-323-3728. Illinois residents call: 800-942-6251. Then after the tone, dial 344.

Scott, Foresman and Company
Electronic Publishing Division
1900 East Lake Avenue Glenview, Illinois 60025

A TRICKY COURTSHIP

BY BARRY BENDAR AND JUDY HEROLD

It's February 1955 and the Sodaville High School gym is decorated with pink cardboard hearts in preparation for the big Valentine's Day dance tomorrow night. All the kids in school are looking forward to going, but two juniors, Kurt and Dede, are especially excited. They've been eyeing each other across the room in chemistry class for months now, and see the dance as their big opportunity to make their secret feelings known.

Everyone knows Dede. She's the most popular girl in the school. She's pretty, self-assured, and flirtatious, and it seems like nearly every guy at Sodaville High has a crush on her. She's never alone; there's always some suitor trailing at her heels. Dede's parents are well aware of their daughter's attractiveness, and perhaps that's why they are especially strict with their popular daughter, and sometimes even snoopy.

Dede's most recent suitor has been Bob, her brother's best friend and a senior at the school. Most girls would love to be handsome Bob's girlfriend, even though he is a terrible dancer. On the surface, cheerleader Dede and football star Bob seem like the perfect couple. But Dede's heart is elsewhere.

Kurt is different from the other boys, and that's why Dede likes him. He works after school and on weekends at a local gas station. Some kids think Kurt's a snob because he's always by himself. But Dede prefers to think of him as an individual, as someone who doesn't need to follow the crowd. And now that Dede has noticed Kurt, her best friend, Clara, has developed a crush on him, too.

A blizzard is blowing outside, but that isn't going to stop any of the kids from going to the dance. The radio says it should clear up the day after the dance. Poor visibility caused Kurt to drive into a snow bank two nights ago, and he hopes to repair his car in time for the dance. The only thing he's dreading is the dancing itself; he's self-conscious and uncoordinated on the dance floor. But he'll suffer a few jitterbugs if it will help him win Dede's heart.

True Love requires two players, preferably of the opposite sex. The male should play the character of Kurt, and the female the character of Dede.

There are nine situations to move through, starting with how to get to the dance. When *True Love* is run, first the male player will be asked how he, as Kurt, should go to the dance. The computer screen will list three numbers, which match choices listed on this page. Once the male player has selected his choice and pressed the RETURN or ENTER key, the screen will clear, and the female, as Dede, will be asked to make her selection. There is only one correct choice for each. If one or both players make the wrong choice, both will be asked to try again before they can move on to the next situation.

If you make the right moves, romance will blossom by the end of the evening. But remember: There are a lot of things to consider in starting any new relationship, and all sorts of blunders can occur to thwart budding love.

The solution will appear in next month's issue.

Had BARRY BENDAR *and* JUDY HEROLD *gone to their own high school dances, they might have met earlier than in college, where, to their surprise, they discovered that they'd attended the same New Jersey high school. Barry is now a programmer/analyst for Bradford National Computer Services and Judy is a freelance writer. They've been going together for four years.*

Base Version (Apple)/*True Love*

```
10 DIM ST$(18),ANS(54),SC(1),A(69):DUR = 20
20 FOR I = 1 TO 18:READ ST$(I):IF ST$(I) = "!" THEN ST$(I) =
ST$(I - 1)
30 NEXT I
40 FOR I = 0 TO 28:READ A:POKE 768 + I,A:NEXT I
50 FOR I = 1 TO 54:READ ANS(I):NEXT I
60 FOR I = 0 TO 69:READ A(I):NEXT I
90 HOME:HTAB 10:FLASH:PRINT "A TRICKY COURTSHIP":CV = 0:GOSUB
 2000:NORMAL
100 COUNT = 1:SC(0) = 1:SC(1) = 1
110 FOR LOOP = 0 TO 8
120 FOR INLP = 0 TO 1
130 HOME
140 IF INLP = 0 THEN WHO$ = "Kurt":S$ = "Boy's":RESP = LOOP *
 6 + 1:GOTO 160
150 WHO$ = "Dede":S$ = "Girl's":RESP = LOOP * 6 + 4
160 PRINT WHO$;":":PRINT
170 R$ = ST$(COUNT):GOSUB 1000
180 FOR OP = RESP TO RESP + 2:PRINT INT(ANS(OP)):NEXT OP:PRIN
T
190 PRINT S$;" choice";:INPUT CHOICE
200 QR = ANS(RESP) - INT(ANS(RESP)):QZ = INT(10 * QR + 0.1):V
 = INT(ANS(QZ + RESP - 1))
210 IF CHOICE <> V THEN SC(INLP) = SC(INLP) + 1:ERFLG = 1
220 IF INLP <> 1 OR ERFLG = 0 THEN 260
230 INLP = 0:ERFLG = 0
240 COUNT = COUNT - 1
250 PRINT:PRINT "SOMEONE GOOFED!":FOR D = 1 TO 2000:NEXT D:GO
TO 130
260 COUNT = COUNT + 1:NEXT INLP:CV = 30:GOSUB 2000:NEXT LOOP
270 HOME
280 FOR I = 0 TO 1
290 IF I = 0 THEN WHO$ = "Kurt":GOTO 310
300 WHO$ = "Dede"
310 IF SC(I) <= 3 THEN R$ = "  You are Uncommonly Cool. Top m
arks in Romance."
320 IF SC(I) > 3 AND SC(I) < 9 THEN R$ = "  In Romance, You'r
e Pretty Cool."
```

1. "I wonder if we'll get married?"
2. Ask her if she's seeing Bob
3. Mention what a lovely night it's been
4. Suggest going for a drive
5. Wait until Dede's alone to ask her to dance
6. Stand by the door
7. Your straight skirt and new blue button-down cashmere sweater
8. Go alone in the family pickup truck
9. Drive your family's Cadillac
10. Your white blazer, red shirt, and slacks
11. Stand by yourself
12. Interrupt and ask Dede to dance
13. Suggest driving gang to Burger Hop for late snack
14. Invite Kurt in for a cup of hot chocolate
15. No
16. Put your arm around her
17. "I think I'm falling in love!"
18. Yes
19. Park in front of her house
20. Mention you have to be home soon
21. Talk to your girlfriends
22. Immediately ask Dede to dance
23. Your black leather jacket, white T-shirt, and jeans
24. Get a ride in friend's new Thunderbird
25. Get ride with brother and Bob
26. Your varsity jacket and corduroys
27. Stand by bleachers with girlfriends
28. Dance with Bob
29. Ask her to get some fresh air
30. Park on nearby Lovers' Peak
31. Ask her to the football game tomorrow
32. Hold her hand
33. "I never thought this would happen!"
34. Maybe
35. Park and walk her to the front door
36. Smile shyly and look the other way
37. Dance with another girl
38. Stand by refreshment table with brother and Bob
39. Your new pink taffeta dress and high heels
40. Take your own Chevy
41. Hop a ride with friends
42. Your red pants and white angora sweater
43. Sit in the bleachers and survey the crowd
44. Talk to Bob but look distracted
45. Lock eyes with Kurt
46. You're nervous, say nothing
47. Ask her to go steady
48. Lean over and kiss her
49. Pretend to ignore Kurt and continue talking
50. Talk to your buddies
51. Suggest going bowling with your best friend and her date
52. Ask her girlfriends if they've seen her
53. Music starts up again, keep dancing
54. Ask Dede's best friend to dance

MODIFICATIONS FOR OTHER COMPUTERS

ADAM/*True Love*

Use the base version, with the following alterations: Omit lines 40, 60, 380, 2000-2070, 5000-5020, and 7000-7070. In line 1000, change 40 to 31. In line 1010, change 39 to 30. Finally, change lines 90, 260, 370, and 390 to read as follows:

```
90 HOME:HTAB 8:PRINT "A TRICKY COURTSHIP":FOR d=1 TO 2500:NEX
T d
260 count=count+1:NEXT inlp:NEXT loop
370 FOR d = 1 TO 2500:NEXT d
390 HOME:HTAB 3:VTAB 10:PRINT "True Love Conquers All!"
```

Atari/*True Love*

Use the base version, with the following alterations: Omit lines 2050 and 5000-5020. In line 1000, change 40 to 38. Finally, change lines 10-60, 90, 130, 170, 250, 270, 370-400, 1010, 1030, 2010-2040, 2060, and 7000-7070 to read as follows:

```
10 DIM ST$(672),S(19),R$(80),ANS(54),A(69),S$(6),WHO$(4),SC(1
):POKE 752,1:VLM=15
20 FOR I=1 TO 18:S(I)=LEN(ST$)+1:READ R$:IF R$="!" THEN ST$(S
(I))=ST$(S(I-1),S(I)-1):GOTO 40
30 ST$(S(I))=R$
40 NEXT I:S(I)=LEN(ST$)+1
50 FOR I=1 TO 54:READ A:ANS(I)=A:NEXT I
60 FOR I=0 TO 69:READ A:A(I)=A:NEXT I
90 PRINT CHR$(125):POSITION 10,0:PRINT "A TRICKY COURTSHIP":C
V=0:GOSUB 2000
130 PRINT CHR$(125)
170 R$=ST$(S(COUNT),S(COUNT+1)-1):GOSUB 1000
250 PRINT :PRINT "SOMEONE GOOFED!":FOR D=1 TO 500:NEXT D:GOTO
 130
270 PRINT CHR$(125)
370 FOR I=1 TO 4:SOUND 0,A(I+65),10,10
380 FOR D=1 TO 100:NEXT D:NEXT I
390 PRINT CHR$(125):POSITION 8,10:PRINT "True Love Conquers A
ll!"
400 CV=0:DN=1:GOSUB 2000:END
1010 FOR L=1 TO 37:IF ASC(R$(L,L))=32 THEN J=L
1030 PRINT R$(1,J):R$=R$(J+1,LEN(R$)):GOTO 1000
2010 FOR X=1 TO RP:SA=CV:FOR Y=1 TO PL:CV=CV+1:VLM=VLM-(0.15*
DN):IF VLM<0 THEN RETURN
2020 SOUND 0,A(CV),10,VLM:SOUND 1,A(CV),10,VLM:SOUND 2,A(CV)+
1,10,VLM
2030 IF INT(CV/2)=CV/2 THEN FOR D=1 TO 30:NEXT D:GOTO 2060
2040 FOR D=1TO 15:NEXT D
2060 NEXT Y:CV=SA:NEXT X:CV=CV+PL+1:IF CV>=65 THEN SOUND 0,0,
0,0:SOUND 1,0,0,0:SOUND 2,0,0,0:RETURN
7000 DATA 4,8,162,81,128,162,81,108,114,108
7010 DATA 2,8,121,60,96,121,60,81,85,81
7020 DATA 2,8,162,81,128,162,81,108,114,108
7030 DATA 1,16,108,72,85,108,121,144,136,128
7040 DATA 121,81,96,121,136,128,136,128
7050 DATA 1,16,162,81,128,162,121,162,114,81
7060 DATA 108,108,121,121,128,128,144,144
7070 DATA 108,53,64,85
```

Commodore 64/*True Love*

Use the base version, with the following alterations: Omit lines 40 and 5000-5020. In line 10, change `DUR=20` to `PRINT CHR$(14)`. Add lines 70-80 and 3000-3020:

```
70 FOR I=0 TO 24:POKE 54272+I,0:NEXT I
80 POKE 54296,15:POKE 54277,57:POKE 54278,128:POKE 53281,10:P
RINT CHR$(144)
3000 HF=INT(N/256):LF=N-(HF*256)
3010 POKE 54273,HF:POKE 54272,LF
3020 POKE 54276,33:RETURN
```

Finally, change lines 90, 130, 250, 270, 370-400, 2020-2050, 4000, and 7000-7070 to read as follows:

```
90 PRINT CHR$(147):PRINT TAB(10) "A TRICKY COURTSHIP":CV=0:GO
SUB 2000
130 PRINT CHR$(147)
250 PRINT:PRINT "SOMEONE GOOFED!":FOR D=1 TO 1000:NEXT D:GOTO
 130
270 PRINT CHR$(147)
370 FOR I=1 TO 4:N=A(I+65):GOSUB 3000
380 FOR D=1 TO 500:NEXT D:POKE 54276,0:NEXT I
390 PRINT CHR$(147):PRINT TAB(255) TAB(248) "True Love Conque
rs All!"
400 CV=0:DN=1:GOSUB 2000:END
2020 N=A(CV):IF DN=1 THEN POKE 54296,INT((70-CV)/4.6)
2030 GOSUB 3000:IF INT(CV/2)=CV/2 THEN FOR D=1 TO 40:NEXT D:G
OTO 2050
2040 FOR D=1 TO 10:NEXT D:IF DN=0 THEN FOR D=1 TO 15:NEXT D
2050 POKE 54276,0
4000 DATA The Valentine's Day Dance is tomorrow. How should y
ou get there?,!
7000 DATA 4,8,1604,3212,2024,1604,3212,2408,2272,2408
7010 DATA 2,8,2144,4288,2700,2144,4288,3212,3032,3212
7020 DATA 2,8,1604,3212,2024,1604,3212,2408,2272,2408
7030 DATA 1,16,2408,3608,3032,2408,2144,1804,1908,2024
7040 DATA 2144,3212,2700,2144,1908,2024,1908,2024
7050 DATA 1,16,1604,3212,2024,1604,2144,1604,2272,3212
7060 DATA 2408,2408,2144,2144,2024,2024,1804,1804
7070 DATA 4816,9632,8096,6064
```

VIC-20 w/8K RAM Cartridge/*True Love*

Use the Commodore 64 modifications, with the following alterations: Omit lines 70, 80, and 3000-3020. In line 10, omit `:PRINT CHR$(14)`. In line 90, change `TAB(10)` to `TAB(2)`. In line 390, change `PRINT TAB(255) TAB(248)` to `PRINT TAB(220)`. In line 1000, change 40 to 22. In line 1010, change 39 to 21. In line 2040, change 10 to 20 and 15 to 30. Finally, change lines 370, 380, 2000-2030, 2060, and 7000-7070 to read as follows:

```
370 POKE 36874,0:POKE 36875,0:POKE 36878,15:FOR I=1 TO 4:POKE
 36876,A(I+65)
380 FOR D=1 TO 300:NEXT D:NEXT I:FOR D=1 TO 500:NEXT D
2000 POKE 36878,15:RP=A(CV):PL=A(CV+1):CV=CV+1
2010 FOR X=1 TO RP:SA=CV:FOR Y=1 TO PL:CV=CV+1:IF DN=1 THEN P
OKE 36878,INT((70-CV)/4.6)
2020 POKE 36875,A(CV):POKE 36876,A(CV):POKE 36874,A(CV)+1
2030 IF INT(CV/2)=CV/2 THEN FOR D=1 TO 100:NEXT D:GOTO 2050
2060 NEXT Y:CV=SA:NEXT X:CV=CV+PL+1:IF CV>=65 THEN POKE 36878
,0:RETURN
7000 DATA 4,8,175,215,191,175,215,201,199,201
7010 DATA 2,8,195,225,207,195,225,215,212,215
7020 DATA 2,8,175,215,191,175,215,201,199,201
7030 DATA 1,16,201,219,212,201,195,183,187,191
7040 DATA 195,215,207,195,187,191,187,191
7050 DATA 1,16,175,215,191,175,195,175,199,215
7060 DATA 201,201,195,195,191,191,183,183
7070 DATA 201,228,223,212
```

IBM PC/*True Love*

Use the base version, with the following alterations: Omit lines 40 and 5000-5020. In lines 130 and 270, change `HOME` to `CLS`. In line 1000, change 40 to 80. Finally, change lines 10, 90, 250, 380, 390, 1010, 2020-2050 , and 7000-7070 to read as follows:

```
10 DIM ST$(20), ANS(54), SC(1), A(70):DR=2:KEY OFF
90 CLS:LOCATE ,31:COLOR 16,7:PRINT " A TRICKY COURTSHIP ": CO
LOR 7,0:CV=0:GOSUB 2000
250 PRINT:PRINT "SOMEONE GOOFED!":FOR D=1 TO 1000:NEXT D:GOTO
 130
380 CLS:LOCATE 12,28:COLOR 16,7:PRINT "True Love Conquers All
!"
390 FOR I=1 TO 4:SOUND A(I+65),10:NEXT I
1010 FOR L= 1 TO 79:IF ASC(MID$(R$,L,1))=32 THEN J=L
2020 IF DN=1 THEN DR=DR-.005
2030 IF INT(CV/2)=CV/2 THEN D=DR*2:GOTO 2050
2040 D=DR
2050 SOUND A(CV),D
7000 DATA 4,8,196,392,247,196,392,293,270,293
7010 DATA 2,8,261,523,329,261,523,392,360,392
7020 DATA 2,8,196,392,247,196,392,293,270,290
7030 DATA 1,16,293,440,360,293,261,220,230,246
7040 DATA 261,392,329,261,230,249,230,249
7050 DATA 1,16,196,392,246,196,261,196,275,392
7060 DATA 293,293,261,261,246,246,220,220
7070 DATA 293,587,493,363
```

TI-99/4A w/TI Extended BASIC/*True Love*

Use the base version, with the following alterations: First, use a double colon (::) instead of a single colon to separate multiple statements on a single numbered program line. So, for example, you would change line 20 to read

```
20 FOR I = 1 TO 18 :: READ ST$(I) :: IF ST$(I) = "!" THEN ST$
(I) = ST$(I - 1)
```

Second, omit lines 40, 380, 2020, and 5000-5020. Third, change HOME to CALL CLEAR in lines 130 and 270. Fourth, change 40 to 28 in line 1000. Finally, change lines 10, 90, 250, 370, 390, 400, 1010, 1030, 2030-2050, and 7000-7070 to read as follows:

```
10 DIM ST$(18),ANS(54),SC(1),A(69) :: VLM=0
90 CALL CLEAR :: PRINT TAB(5);"A TRICKY COURTSHIP" :: CV=0 ::
 GOSUB 2000
250 PRINT :: PRINT "SOMEONE GOOFED!" :: FOR D=1 TO 300 :: NEX
T D :: GOTO 130
370 FOR I=1 TO 4 :: CALL SOUND(600,A(I+65),0) :: NEXT I
390 CALL CLEAR :: PRINT TAB(255):TAB(255):"True Love Conquers
 All!"
400 CV=0 :: DN=1 :: GOSUB 2000 :: END
1010 FOR L=1 TO 27 :: IF ASC(SEG$(R$,L,1))=32 THEN J=L
1030 PRINT SEG$(R$,1,J-1) :: R$=SEG$(R$,J+1,LEN(R$)-J) :: GOT
O 1000
2030 IF INT(CV/2)=CV/2 THEN DUR=180 :: GOTO 2050
2040 DUR=90
2050 VLM=VLM+(.3*DN) :: CALL SOUND(DUR,A(CV),VLM)
7000 DATA 4,8,196,392,247,196,392,294,277,294
7010 DATA 2,8,262,523,330,262,523,392,370,392
7020 DATA 2,8,196,392,247,196,392,294,277,294
7030 DATA 1,16,294,440,370,294,262,220,233,247
7040 DATA 262,392,330,262,233,247,233,247
7050 DATA 1,16,196,392,247,196,262,196,277,392
7060 DATA 294,294,262,262,247,247,220,220
7070 DATA 587,1175,988,740
```

TRS-80 Color Computer/*True Love*

Use the base version, with the following alterations: Omit lines 40, 380, 2020, and 5000-5020. In lines 130 and 270, change HOME to CLS. In line 1000, change 40 to 32. In line 1010, change 39 to 31. Finally, change lines 10, 90, 370, 390, 400, 2030-2050, and 7000-7070 to read as follows:

```
10 DIM ST$(18),ANS(54),SC(1),A(69)
90 CLS:PRINT @ 7,"A TRICKY COURTSHIP":CV=0:GOSUB 2000
370 FOR I=1 TO 4:SOUND A(I+65),10:NEXT I
390 CLS:PRINT @ 228,"True Love Conquers All!"
400 CV=0:GOSUB 2000:GOTO 400
2030 IF INT(CV/2)=CV/2 THEN DR=2:GOTO 2050
2040 DR=1
2050 SOUND A(CV),DR
7000 DATA 4,8,32,147,78,32,147,108,99,108
7010 DATA 2,8,89,176,125,89,176,147,140,147
7020 DATA 2,8,32,147,78,32,147,108,99,108
7030 DATA 1,16,108,159,140,108,89,58,69,78
7040 DATA 89,147,125,89,69,78,69,78
7050 DATA 1,16,32,147,78,32,89,32,99,147
7060 DATA 108,108,89,89,78,78,58,58
7070 DATA 108,185,170,140
```

TRS-80 Models I & III/*True Love*

Use the base version, with the following alterations: Omit lines 40, 60, 380, 2000-2070, 5000-5020, and 7000-7070. In line 1000, change 40 to 64. In line 1010, change 39 to 63. In lines 130 and 270, change HOME to CLS. Finally, change lines 10, 90, 260, 370, and 390 to read as follows:

```
10 CLEAR 1000:DIM ST$(18),ANS(54),SC(1)
90 CLS:PRINT @ 25,"A TRICKY COURTSHIP":FOR D=1 TO 1000:NEXT D
260 COUNT=COUNT+1:NEXT INLP:NEXT LOOP
370 FOR D=1 TO 1000:NEXT D
390 CLS:PRINT @ 466, "True Love Conquers All!"
```

TRS-80 Model 4/*True Love*

Use the Model III version, with the following modifications: In line 90, change 25 to 31. In line 390, change 466 to 587. In line 1000, change 64 to 80. Finally, in line 1010, change 63 to 79.

GAMES

Title Manufacturer Price	Brief Description	Hardware/ Equipment Required	Backup Policy	Ratings O	D	EH	GQ	EU	V
ARACHNOID United Microware Industries 3503 Temple Ave., Suite C Pomona, CA 91768 (714) 594-1351 $44.95 © 1983	Defend your spider's nest eggs from marauding ants and wasps in shoot-'em-up with several variations, including a time factor. Suitable for ages 8 and up. —DELSON	Commodore VIC-20 (cartridge); joystick required	Defective disks replaced free w/ in 90 days; nominal fee if user-damaged.	★ ★ ★	★ ★	★ ★ ★	★ ★	A	★ ★ ★
BROADSIDES Strategic Simulations 883 Stierlin Rd. Building A-200 Mountain View, CA 95827 (415) 964-1353 $39.95 © 1983	Command a preprogrammed warship or one you've built yourself. Set sail to defeat computer or human opponent. Ages 8+ will enjoy arcade action; 12+ can utilize more complex tactical elements.† —DELSON	Apple II/II plus/IIe/III w/ emulator, 48K (disk)	Defective or user-damaged disks replaced free w/in 30 days; $10 fee thereafter.	★ ★ ★ ★	★ ★ ★ ★	★ ★ ★	★ ★	A	★ ★ ★ ★
CAPTURE THE FLAG Sirius Software 10364 Rockingham Dr. Sacramento, CA 95827 (916) 366-1195 $39.95 © 1983	Take on computer or human opponent and move through 3-D, high-resolution labyrinth in search of flag. Use map or view of immediate surroundings and evade your pursuer. —DELSON	Atari 400/800/1200, 48K (disk); also available for Commodore VIC-20, (cartridge); joystick required for 2 players	Defective material replaced free w/ in 30 days; $5 fee if user-damaged.	★ ★ ★	★ ★	★ ★	★ ★ ★	A	★ ★ ★
DEFENDER ON TRI NUFEKOP P.O. Box 156 Shady Cove, OR 97539 (503) 878-2113 $12.95 © 1983	Maneuver spaceship through hazardous maze to rescue scientist. Play against the clock, avoiding traps, eliminating obstacles. Great fun and excitement for ages 8+. —DELSON	Commodore VIC-20, 3K (cassette); joystick required	Defective material replaced free w/in 90 days; user makes backup copy.	★ ★ ★	★ ★	★ ★	★ ★	A	★ ★ ★ ★
FORTRESS OF THE WITCH KING Avalon Hill Game Co. 4517 Hartford Rd. Baltimore, MD 21214 (301) 254-5300 $25 © 1983	Journey across fantasy landscape with a troop of adventurers, searching for your mighty foe, the Witch King. Many levels of play and 20 map boards, great for ages 10+.† —DELSON	Apple II/II plus/IIe, 48K (disk)	Defective disks replaced free; user makes backup copy.	★ ★ ★	★ ★ ★ ★	★ ★ ★	★ ★	A	★ ★ ★ ★
MICROSOFT DECATHLON IBM P.O. Box 1328 Boca Raton, FL 33432 (305) 241-7614 $29.95 (Apple) $35 (IBM) © 1983	Compete with up to six armchair athletes in races, hurdles, shot puts, discus and javelin throws, and pole vaults in exciting game for ages 8 and up.† —DELSON	IBM PC, 64K (disk); also available for Apple II/II plus/IIe, 48K (disk); IBM PC requires color card	Defective disks replaced free w/ in 90 days; user makes backup copy.	★ ★ ★ ★	★ ★ ★	★ ★ ★	★ ★ ★	A	★ ★ ★ ★
M.U.L.E. Electronic Arts 255 Campus Dr. San Mateo, CA 94403 (415) 571-7171 $40 © 1983	Claim land, mine for intergalactic minerals, grow food and energy. Bid and bargain, trade and sell surplus in unusual economic game for ages 11 and up.† —DELSON	Atari 400/800/1200, 48K (disk); also available for Commodore 64 (disk); joystick required	Defective disks replaced free w/ in 90 days; backup copy available for $7.50.	★ ★ ★	★ ★ ★	★ ★	★ ★	A	★ ★ ★
RABBIT TRAIL Funware 230 E. Caribbean Dr. Sunnyvale, CA 94089 (408) 745-1655 $40 © 1983	Hop along bunny path, eating carrots, avoiding traps, weasels, hawks, and cars in surprisingly challenging game playable by all ages 6+. —DELSON	TI-99/4A (cartridge); joystick required	Defective material replaced free w/ in 90 days.	★ ★ ★	★ ★	★ ★	★ ★	A	★ ★
TITAN EMPIRE Muse Software 347 N. Charles St. Baltimore, MD 21201 (301) 659-7212 $34.95 © 1983	Criss-cross solar system, doing away with evildoers in game with complex play system. Provides tactical, strategic challenge best-suited for ages 13+.† —DELSON	Apple II/II plus/IIe, 48K (disk)	Defective material replaced free; $10 if user-damaged.	★ ★ ★	★ ★ ★	★ ★	★ ★	A	★ ★ ★
ZEPPELIN RESCUE Computer Software Assoc. The Silk Mill, 44 Oak St. Newton Upper Falls, MA 02164 (617) 527-7510 $19.95 (cassette) $24.95 (disk) © 1983	Maneuver clumsy blimp over city and rescue endangered citizens from tops of buildings, beneath bridges. Simple but compelling game requires nerves of steel and delicate touch. —DELSON	Commodore 64 (disk or cassette); joysticks required	Defective disks replaced free w/ in 90 days.	★ ★ ★	★ ★ ★	★ ★	★ ★	A	★ ★ ★

RATINGS KEY **O** Overall performance; **D** Documentation; **EH** Error handling; **GQ** Graphics quality; **EU** Ease of use; **V** Value for money; ★ Poor; ★★ Average; ★★★ Good; ★★★★ Excellent; n/a Not applicable; E Easy; A Average; D Difficult; † Longer review follows chart