Journal of the ICCE Special Interest Group for Logo-Using Educators

LOGO EXCHANGE

SEPTEMBER 1987

VOLUME 6 NUMBER 1

International Council for Computers in Education

ICCE Publications

University of Virginia
Summer Logo Fellowship Program
1987 - 1988

What is the UVA Summer Logo Fellowship Program?

An opportunity for educators to come to the University of Virginia for a three week summer period of individual Logo related study.

What are the objectives of the Logo Fellowship program?

The objectives are to:

Encourage the development of innovative Logo projects in all grade levels and subject areas

Disseminate Logo projects produced by the fellowship program

Provide fellowship recipients with time and resources for study, work, and growth

When does the fellowship program operate?

The next fellowship period of study will begin in the summer of 1988. The application cycle for this period begins in October 1987.

How is the fellowship program supported?

The program is made possible by a series of grants from Logo Computer Systems Incorporated (LCSI) to the University of Virginia. From these grants, support for travel, room, board, graduate credit hours, and a cash award are provided to each fellowship recipient.

Who may apply for the UVA Summer Logo Fellowship PRogram?

Teachers of K-12 students, college and university teachers,and teacher educators who have used Logo on a regular basis for the past year are eligible to participate.

What other qualifications are important?

Fellowship recipients must possess exceptional communication skills and initiative, be able to work well with others, and be highly motivated about the possibilities for using Logo in education.

What is the application process?

Applicants should write immediately to the University of Virginia Summer Logo Fellowship Program, Curry School of Education, Ruffner Hall, Charlottesville, VA 22901, or call (804) 924-7471, requesting the application forms. Completed applications are to be returned to the University of Virginia before December 21, 1987. A selection committee will evaluate the applications and choose a group of finalists, from which the three fellowship recipients for 1988 will be named.

What are the fellowship expectations?

A Logo fellowship recipient is expected to plan and begin carrying out a self-designed Logo-related project during the fellowship period. Tom Lough, Glen Bull, Judi Harris, and others will serve as consultants to each fellowship recipient. At the conclusion of the summer study period, the recipient would be expected to submit a reasonable plan for completing the project, and for dissemination of the project.

What is a typical fellowship project?

Projects are conceived and planned by the fellowship candidates themselves. Typical projects could include innovative Logo units related to particular subject areas, teaching or training modules, informal educational research, and course development. Projects carried out under the current series of grants must be based on LCSI Logo products.

Publishers of The Computing Teacher journal.

Dear Logo Exchange Subscriber:

Enclosed is the Spetember issue of *Logo Exchange*. Thank you for your patience!

The International Council for Computers in Education (ICCE) has recently taken over publication of *Logo Exchange* from Meckler Publishing Corporation. Please bear with us as we make the necessary adjustments required of the aquisition.

Because we were not supplied with pertinent subscription information in a timely manner, we got off to a late start on the fall issues. We anticipate being back on track by the first of the year. We apologize for this convenience.

You will notice that Sharon Burrowes will be working in conjunction with Tom Lough as editor-in-chief. Please note also the information on the East Coast Logo Conference.

ICCE is the leading U.S. and international professional organization for computer educators at the precollege level. It is non-profit, supported by more than 50 organizations of computer-using educators worldwide. These organizations are statewide or region-wide in scope, averaging 500 members each.

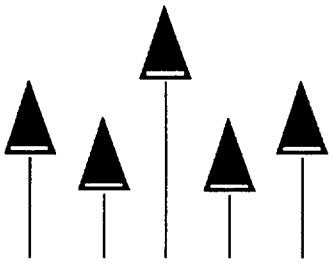
We look forward to a successful year. Thank you again for your patience during the change-over.

Sincerely,

Kathleen M. Geygan

Kathleen M. Geygan
Marketing Director, ICCE

University of Oregon 1787 Agate St. Eugene, OR 97403 USA 503/686-4414



LOGO EXCHANGE

VOLUME 6 NUMBER 1 Journal of the ICCE Special Interest Group for Logo-Using Educators SEPTEMBER 1987

Logo Exchange is the journal of the International Council for Computers in Education Special Interest Group for Logo-using Educators (SIGLogo). It is published monthly September through May by ICCE, University of Oregon, 1787 Agate Street, Eugene, OR 97403-9905, USA. S[redacted]

POSTMASTER: Send address changes to Logo Exchange, University of Oregon, 1787 Agate Street, Eugene, OR 97403-9905, USA.

CONTENTS

From the Editors
Sharon Burrows and Tom Lough 2
SIG News and Conference Announcements 4
IF...THEN
Nancy Parks Sopp 5
Star Patterns: An Alternative Approach to Angles
Reinhold Wappler 7
Teacher Feature
Rebecca Poplin 10
Listful Thinking — Writing by Formula
Glen L. Bull and Gina L. Bull 12
Logo LinX — Blank Endorsement
Judi Harris 16
Printing Long Lines in Apple Logo II 19
LXionary
Bill Craig 20
PenPoints — Logo Book Review with ASTROLUG
Craig Hodgson and Peter Rawitsch 21
ObjectLogo: A Review
Robs Muir 23
Jacques and Elsie
Linda Sherman 26
IntLXual Challenge — Fractal Frolic
Robs Muir 27
International Logo News
Dennis Harper 29
LogoPals
Barbara Randolph 31

SIGLogo Membership
(includes the Logo Exchange)

	U.S.	NON-U.S.
ICCE Member	24.95	29.95
Non-ICCE Member	29.95	34.95

Send membership dues to ICCE. Add $2.50 for processing if payment does not accompany your dues. VISA and Mastercard accepted.

Dear Readers,

When Posy and I started The National LogoExchange newsletter five years ago, we hoped to provide teachers with a means for sharing useful Logo information. Beginning as an 8-page dot-matrix publication, the NLX carried a monthly offering of articles written by teachers and columns written by Steve Tipps and Glen Bull. What an exciting time it was!

Soon, the newsletter graduated to a letter-quality printer. Other contributors joined the effort. Robs Muir and Griff Wigley were the first of nearly two dozen people who eventually became involved with the publication.

Later, a laser printer provided another "upgrade" to the typeset look, with 16-plus pages of Logo information each month. With Dennis Harper, I started the bi-monthly International Logo Exchange newsletter, to respond to the needs of educators wishing to stay informed of Logo developments and use on a global level. Outstanding field editors on each continent reported on "happenings" in their area. For the first time, it was possible to get a sense of the "big picture" of Logo around the world.

Meckler Publishing acquired the NLX and the ILX in 1986 and consolidated them into the Logo Exchange magazine, expanding the offering to 32-plus pages each month.

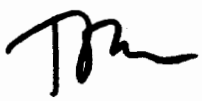
As this publication enters its sixth year, even more changes are evident. It has a wonderful new home with the International Council for Computers in Education (ICCE). It is now associated directly with a Logo organization, with membership open to all. Even more people now have the opportunity to be involved with the writing, editing, and publishing.

In that regard, I invite each of you to consider making a contribution to the LX. Perhaps an article, a short paragraph or two of your favorite Logo tip, or a printout of a student project. As Logo enters what some are calling its "maturation" stage, it will be even more important than ever before to continue to share information and ideas with each other. Let's work to make the Logo Exchange better than ever!

Five years is enough; it is time for me to step aside. With a mixture of nostalgia and anticipation, I pass the torch to the ICCE and the new staff of the Logo Exchange. Sharon Burrowes, a long-time Exchange contributor, has agreed to be editor. Sharon, the present contributors, and the volunteers being added will bring a much needed dimension of breadth to the publication. I plan to continue my writing on a more or less regular basis.

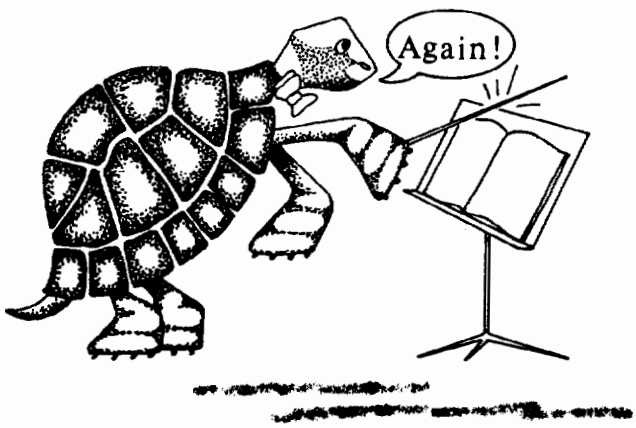
When I think back on the past five short years, I am amazed, humbled, and proud. Steve Tipps and Glen Bull each have written nearly fifty significant Logo pieces. The NLX, ILX, and LX combined have published and distributed nearly six hundred Logo articles! The staff of contributors has become like a second family to me. And you, dear readers, became my extended family. What a pleasure it has been to meet you at conferences and to receive mail and telephone calls from you. From the bottom of my heart, I thank each one of you.

As always,



FD 100!!

Tom Lough
Founding Editor
PO Box 5341
Charlottesville, VA 22905
CompuServe 70020,223



From the Editor

By Sharon Burrowes

Dear Readers, both old and new:

I want to take this opportunity to welcome you to the "new" Logo Exchange. You will find only a couple of changes in this first issue of LX; many familiar "faces" are back providing you with classroom ideas and new challenges.

As the new editor, let me begin by telling you how I managed to find myself in front of this keyboard today. My story begins shortly after the East Coast Logo Fair in March of 1987. Since I am a member of the Board of Directors of the International Council for Computers in Education (ICCE), several members of the Logo community approached me about the possibility of creating a Special Interest Group (SIG) for logo The time seemed ripe for a professional organization focused on Logo using educators.

After discussing the appropriate process with ICCE, we began to circulate petitions for establishment of a SIGLogo throughout the Logo community. (Perhaps you even received one!) The petitions began to fill my mailbox almost immediately; it became clear that indeed there was sufficient interest in the Logo community to create a Logo SIG. At the June board meeting held at NECC (National Educational Computing Conference), the ICCE Board of Directors voted to establish SIGLogo, and I agreed to be newsletter editor.

Later in the summer, a phone call came from ICCE. They were acquiring the Logo Exchange from Meckler Publishing. Clearly LX would be appropriate as a journal for the SIGLogo. Since I had agreed to edit the SIGLogo newsletter and Tom was stepping down as LX editor, how would I like to edit LX?. After the initial ...GULP! .. I agreed. During the rest of a very hectic summer, I spent many hours thinking about how LX and ICCE can benefit each other in the years to come.

If you are not familiar with ICCE, let me introduce you. ICCE is a professional organization of primarily pre-college computer using educators. ICCE publishes the highly respected journal, *The Computing Teacher*. In addition, they support publication of a number of smaller newsletters and journals, publish a variety of materials for all aspects of computers in education, sponsor a number of SIG's, and are intimately involved with the annual NECC conference.

As a subscriber to the LX, you automatically become a member of the SIGLogo. If you are not a member of ICCE, I would encourage you to join. Not only does it reduce the cost of your SIG membership (and thus LX), but your subscription to *The Computing Teacher* gives you some excellent Logo articles each month as well as a window on the world of educational computing as a whole.

Officially establishing the SIGLogo and putting together the first new issue of LX is just the first step. There is much yet to be done. Both the SIGLogo organization and LX need your help. What can you do? Many things.

- •Write an article for LX
- •Volunteer to review articles for LX
- •Become active in the SIGLogo
- •Volunteer to be an officer
- •Help with development of the constitution and by-law
- •Help establish SIGLogo meetings at local conferences
- •Become active in your local Logo group and send news of their activities to ICCE/LX.
- •Communicate your opinions about the content of LX and the activities of the SIGLogo.

With your help, LX will continue to grow in directions helpful to an even wider segment of the Logo community and the SIGLogo will establish links among local Logo groups. I look forward to hearing from you or seeing you at regional or national conferences.

Sharon Burrowes, Editor
ICCE-SIGLogo
University of Oregon
1787 Agate Street
Eugene, Oregon 97403
CompuServe 73007,1645

Cover: Patty Turning and Liz Kiefer, Parkview Elementary, Wooster, OH

Patty and Liz have been working with LogoWriter for two years. They had completed the LogoWriter Activity Cards and were creating their own project. The graphics are only a part of the project. The entire program has the basketball bounced and shot through the basket with a musical finale. The girls claim it isn't finished, but then most LogoWriter projects rarely are.

SIGNews and Conference Announcements

Great Lakes/ East Coast Logo Conference

The Great Lakes/ East Coast Logo Conferencewill be held on May 6-7, 1988 with preconference workshops on May 5. This conference, held at the Stouffer Tower City Hotel in Cleveland, OH will feature numerous well-known speakers: Seymour Papert, Dan and Molly Watt, Michael Tempel, Tom Lough, Brian Silverman, Steve Ocko, Mike Battista, Doug Clements, Alison Birch and many more.

Presentations that describe university, district, and school based Logo projects, address issues of equitable use of computers and present uses of Logo beyond the introductory level are particularly desired. Deadline for submission of proposals is January 15, 1988.

For more information on submitting proposals, contact Sharon Burrowes, L.C.S.I., 330 W. 58th, Suite 5M, New York, NY 10019-2925.

National Educational Computing Conference (NECC) '88

Loews Anatole Hotel, Dallas, TX, June 15 - 17, 1988. For more information contact Jim Poirot, NECC '88, Department of Computer Education and Cognitive Systems, North Texas State University, P.O. Box 13886, NT Station, Denton, Texas, 76203-3886.

Cover Art

Do you have interesting pictures, patterns, or designs that either you or your students have produced using Logo? This year the cover of the *Logo Exchange* will feature pictures generated by LX readers using any version of Logo. Submit your "entries" both on disk and with camera ready copy to Sharon Burrowes.

We need to hear from you. Tell us about your local Logo users group- What have you been doing? What activities are you planning? What conferences in your area will feature Logo presentions and/or workshops?

Send your news items to

Sharon Burrowes
ICCE-SIGLogo
University of Oregon
1787 Agate Street
Eugene, Oregon 97403-9905

We're looking forward to hearing from you!!!

IF. . .THEN

by
Nancy Parks Sopp

Six years ago I spent my entire year's teaching supply budget on a robot turtle from M.I.T. and the Logo language. I had just received a brand-new Apple II+ for my junior high resource room and didn't know what to do with it. A trusted colleague told me that Logo was just the thing for my learning disabled students. It was. I wonder now if she suspected the impact that a floppy disk and a little robot would have on my teaching style.

My students and I began learning Logo together. They wrote programs to make the robot turtle move about our classroom on its long cord, tooting its horn and flashing its eyes—red lights soldered to a circuit board. The high point came when we learned to program it to respond to feedback from its touch sensors.

Here is a sample program:

```
TO MOVE
IF FTOUCH? THEN BACK 20 RIGHT 90
FD 20
MOVE
END
```

The MOVE procedure makes the turtle check to determine if its front touch sensor is activated by someone touching its shell or the turtle running into something.

If it is not, the turtle moves forward 20 steps (about 6 inches) and checks again. If the sensor is activated, the turtle rolls backward, turns right 90 degrees, and moves forward once again. The turtle keeps moving about the room this way until someone interrupts the program from the keyboard with CONTROL-G.

My students and I learned Logo the same way the robot behaved in its MOVE procedure. We went slowly through the chapters in the M.I.T. Logo technical manual and Hal Abelson's book, Logo for the Apple II. IF we bumped into something that intrigued us, THEN we stopped our sequence of readings, changed course, and moved forward in a different direction.

That first year, we wrote a text adventure story, spent hours creating recursive designs, and programmed our Apple II+ to play the role of a sarcastic psychoanalyst.

The students wrote programs for the robot off and on throughout the year. Sometimes they made errors in writing the IF...THEN statement, so that when the robot ran into something, it would ignore the input from its touch sensors and spin its wheels, desperately trying to move forward, oblivious to the fact that it was making no progress.

This IF...THEN teaching and learning style was a big change from my former practice, in which I decided what course our learning would take and then moved forward relentlessly, without regard for the students' interests. In learning Logo, I gave students a voice in what they learned.

For instance, I came upon the branching adventure story idea in *The Computing Teacher*[1]. I proposed it to the students, thinking it would provide much-needed practice on language arts skills. They agreed to write one. Mark, testing the limits of my new teaching style, said "Let's call it 'The Day the Teacher Dropped Dead'."

The title stuck, and these easily bored students stuck with their story until it had nine endings, several of which featured the untimely deaths of their favorite teachers. The project gave Jason, the class clown, an appropriate outlet for his humor. Mark, a disheveled boy who couldn't organize himself enough to bring his glasses, paper and pencil to class, helped me organize the story segments. Tara dropped her look of perpetual boredom and spent long hours helping Linda, who could barely read or write, type her story segments.

The IF...THEN style of learning carried over into our time away from the computer. For example, Chris had been spinning his wheels in our spelling lessons, making no progress learning new words. He asked to take over a project that a student in the gifted/talented program had given up on—repairing a motorcycle engine. He had to order parts for it, so he learned to spell magneto and carburetor, thus moving forward in learning spelling, but moving in a different direction.

The following year I got a new job. I packed the robot in foam peanuts and carried it to one of the two new computer labs which I was to supervise. I

had no students of my own, being responsible instead for teaching other teachers' students everything from single-digit subtraction to spreadsheets. Overwhelmed by my new responsibilities, I neglected to unpack the turtle and forgot the IF...THEN style of learning and teaching that had worked so well.

A teacher at the nearby elementary school, Mrs. Sanz, requested Logo instruction. Every week or two she shepherded her third graders across the street to my high school computer lab, where I gave them a Logo lesson.

Several years into my new job, I was teaching Mrs. Sanz' latest crop of third graders. It was the last week in October. Tromp..tromp... I heard their heavy boots coming down the hall. With big smiles on their rosy faces, they filed into the lab. Mrs. Sanz thrust a sheaf of smudged papers into my hands. I glanced at the first one—"Dear Mrs. Sopp..." was all I read before setting them on my desk to read later. I proceeded with my lesson on pentagons and hexagons, using the same lesson plans that I had used the previous two years. After they left, I read the letters. Halfway through the stack, I read one that acted like a CONTROL-G to interrupt my carefully sequenced lesson plans.

> Dear Mrs. Sopp,
>
> Thank you for teaching us Logo. I hope you can teach us how to make a pumpkin. And a jack-o-lantern too. I hope you do Mrs. Sopp.
>
> Your student,
>
> Jana

It was nearly Halloween; of course they were interested in pumpkins! I could have programmed a pumpkin shape and a big toothy grin, and they could have programmed the triangle eyes themselves! It would have been a wonderful lesson, incorporating both their interest in the holiday and my goal to have them learn more Logo.

I had not taken their interests into account. I had been teaching Logo from a predetermined plan instead of allowing students' interests to determine the course for our learning.

I had been acting like the robot turtle with an error in its IF. . .THEN statement, ignoring all input from its touch sensors and spinning its wheels, trying to move forward and oblivious to the fact that it wasn't accomplishing its goal.

What is wrong with ignoring the students' interests? They still enjoyed their Logo lessons. But they had glimpsed its potential as a tool they could use to create whatever they wished, for they asked for pumpkins, rockets, and spaceships.

The students had dared to bring an agenda to the computer lab. By ignoring it, I was teaching them that teachers are the ones who decide what is to be learned.

In the months after those letters, I modified my plans according to the students' and Mrs. Sanz' interests. The next time I saw the students we constructed Logo rockets and spaceships. When two girls chattered to me about the costumes their mothers were making for their Thanksgiving play, I remembered a Thanksgiving Logo activity from *Teaching and Computers*[2] On their next visit to the lab they were pilgrims, planting Logo corn and building Logo cabins.

At the end of January I programmed Logo lace, hearts, and the alphabet in large letters[3], and the students made fancy Valentines for mothers and friends. In March, Mrs. Sanz told me of her struggles to introduce fractions to her students. I paged through past issues of *Logo Exchange* and came up with a fractions lesson.

I recently unpacked the robot turtle. It now has a decided list, veering off to the right when it's supposed to move straight forward. But it still responds just fine to IF...THEN statements. And now, most of the time, so do I.

1 "The Logo Center," *TCT*, Vol. 10 No. 5, January 1983 p. 66-71

2 "The Misadventures of Francis Billington," by Candy Colborn, *Teaching and Computers*, p. 20-23, Vol. 4, No. 3, Nov./Dec. 1986

3 "National Logo Exchange Apple Logo Alphabet Procedures," *LX*, December 1982

Nancy Parks Sopp is the learning disabilities specialist for grades one - three, Woodriver Elementary School, Fairbanks, AK.

Star Patterns: An Alternative Approach to Angles

by Reinhold Wappler

Start Logo and type RIGHT 60. Then ask your students:

Show me the angle.....

Or type FD 40 RT 60.

Show me the angle.....

Or type FD 40 RT 60 FD 40 and again ask:

Show me the angle.....

"Conventional" Logo experience does not impart a conventional sense of angle to elementary level children. None of the above show a textbook angle you would draw if asked to show what 60 degrees looked like.

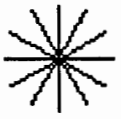
As our Logo-trained children encounter the standard geometry of regular polygons and irregular closed shapes in 5th and 6th grades, we have been struck by how little the two disciplines, as taught, support one another. Several studies have noted that Logo-trained children have a procedural conception of turns and angles quite different from the static geometric notions held by more traditionally instructed kids. Both notions are valuable. It should be possible to adjust the usual approach to Logo so as to familiarize students *simultaneously* with *both* procedural and textbook notions of what angles are. CLIME (Council for Logo in Mathematics Education; see below) member Doug Clements points out, in the April '87 issue of the *Logo Exchange,* that one of Piaget's strongest findings was that children's sense of geometric form is based *on actions they perform on objects*. For this reason we chose early in our program to teach students to use angles in their familiar form to draw shapes like *stars*, and leave regular geometric shapes for later. We have found that very simple procedures yield a rich harvest.

A good starting point is a classroom exercise using this procedure:

```
TO STAR :ANGLE
FD 40
BK 40
RT :ANGLE
STAR :ANGLE
END
```

Type STAR 10, and when it finishes winding out *a full circle* of 10-degree angles *all the way around,* stop and clear the screen. Type

STAR 30...



This one is suitable for counting the arms or the angles. Ask a student to count the 30 degree angles, and several others to trace with a finger the two arms that bound each 30 degree angle. Suggest the multiplication of 30 by 12 to see how many degrees are needed to turn all the way around. Then clear screen and type.STAR 36. This produces ten angles and arms to be counted and traced. Again, multiplying 36 by 12 results in 360.

Continue with the principle angles for 8, 6, 5, 4, 3, and 2-armed stars. Then challenge your class to make a 7, 9, or 11 armed star and work toward understanding of division of 360 by the number of wanted arms. Stress that an input to STAR of 360/11 works just as well as 32.767 and is much easier.

Now draw a sequence of STARs starting with STAR 1, STAR 2, STAR 3, etc., on out as far as you can retain interest. Play with the process the first time; next time have the kids prepared with data collection sheets. This simple exercise can easily consume 2 or more class periods. It can yield pleasure and benefit for grades 5 thru 12, since it involves prediction, counting, comparing, tabular data collec-

tion, prime numbers, calculator exercises, mental arithmetic, division, remainders, factoring, prime factors, faction manipulations, patterns, modulos, formulas, functions, and much more.

Included is a copyable sheet of stars which you may find useful in the version of the exercise you evolve. A tabular form is shown on which students can collect observations as this process goes on.

Kids always want to complete the orbit, at which point "retrace" may occur, with no further filling in, or it may not occur, producing another "orbit" of arms, and sometimes another and another. The process always has an endpoint in a new pattern or in one of the previous patterns on the chart. (One is missing: which?) The basic game is to run STAR with some number input and call for predictions of the endpoint pattern, and the number of orbits that will occur before retrace. Some are obvious, others are tricky.

To help determine if a pattern is an 8-degree or a 10-degree pattern, engage the kids in a writing a procedure to place a comparison star to the right of the test star, as suggested in the drawing below. Then can place another on the left. Direct visual comparisons are made easy.

Calculators may be used to figure number of arms, but don't let calculators be the enemy of mental arithmetic. Since calculators do not figure remainders, it is a challenge to figure out how to convert decimals to remainders. Alternatively, Logo programs can be written to output remainders. Try to write one not using the primitive REMAINDER.

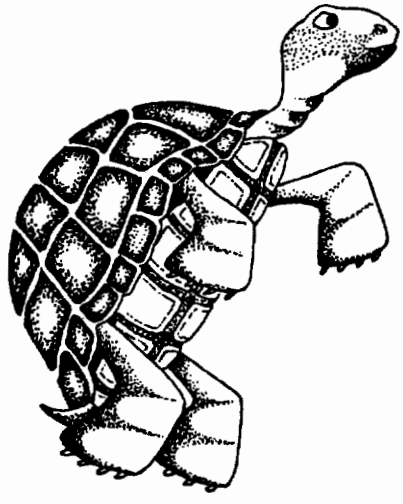
The exercise gets more difficult with larger inputs. Above 360 the notion of modulo is introduced. Use inputs like STAR 2397, and depending on the grade level you can ask for a formula to determine the end point pattern, the number of arms, or the number of orbits required for retrace.

Add a "wait until keypress" line to the STAR procedure such as MAKE "K RC and at the completion of one orbit of, say, STAR 19, ask a student to point to where the next arm will be drawn, and the next and the next, and finally, why?

The paragraph you just read will mean little to you until you sit down, write the procedures, and run them. This is, above all, a dynamic non-verbal process, uniquely demonstrative of the utility of a computer and Logo in your classroom. Typically, our effort to give kids a good sense of what angles are has led us along many mathematical trails. Enjoy!

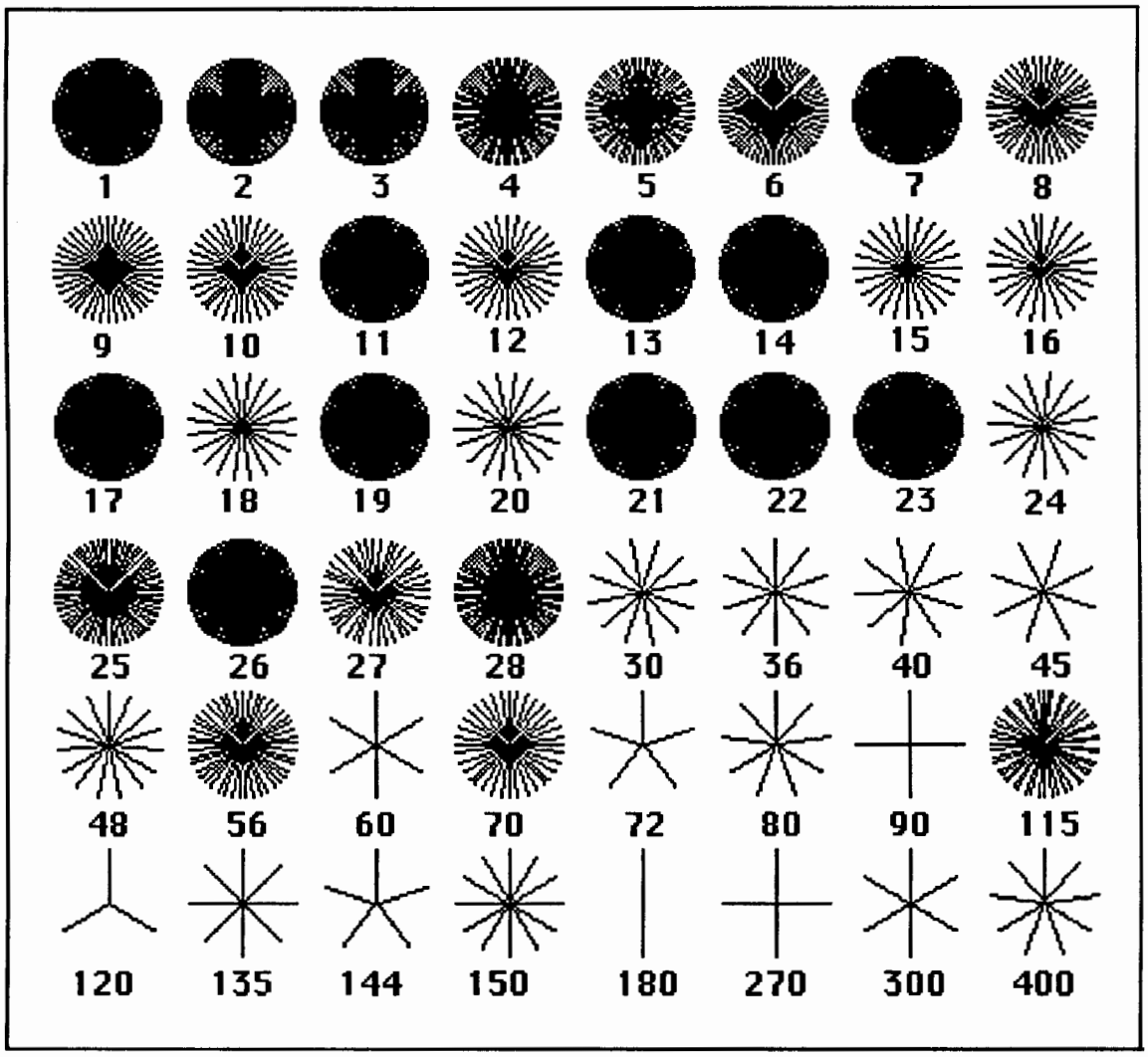
Reinhold Wappler, 252 Carter Street, New Canaan, CT 06840.

This article previously published in the Junk, 1987 CLIME Newsletter. For information on becoming a member of CLIME (the Council for Logo in Mathematics Education), send a self-addressed stamped envelope to Ihor Charischak, Editor, 445 Broadway, Hastings on Hudson, New York 10706.



Patterns produced by Procedure STAR

Numbers below pattern are inputs to STAR



Notes:

```
TO STAR :ANGLE
FD 40
BK 40
RT :ANGLE
STAR :ANGLE
END
```

Below is a suggested form for a table for data collection.

Measure of Angle	Sketch of Angle	Number of Arms in one Orbit	Arms times Angle	Retrace? Yes/No?	Orbits to final pattern	Arms in final pattern	Remainder $\frac{360}{\text{Angle}}$

Teacher Feature

by Rebecca Poplin
Featuring: Mary Upton

A Jeffersonian Logo Experience

Imagine staying in a room designed by Thomas Jefferson on The Lawn of the University of Virginia and working with Logo at the same time! That is what Mary Upton of Lubbock, Texas, experienced this summer. Mary was one of the two Logo Fellowship recipients in a program sponsored by the UVA Curry School of Education and Logo Computer Systems, Incorporated. She was chosen on a competitive basis to spend a three-week period of residence at UVA with all expenses paid to work on a self-designed LogoWriter project.

The fellowship program was coordinated by Tom Lough and Glen Bull. Judi Harris served as a LogoWriter consultant to the fellowship recipients.

Mary proposed to create Logowriter tool pages that will allow children to put together data tables and graphs. As they put the graphs together they will learn how to read and interpret them. The project includes instructional pages which the child may use and tool pages for bar, line, and picture graphs. Children will build graphs from surveys or topics they are studying. When Mary finishes, the materials she has produced will be available in the public domain.

The Groundwork

Mary's interest in Logo developed about 5 years ago when she began taking staff development courses with Bob Knight, computer coordinator in Lubbock. After 150 hours of training, Mary became a computer technologist for the Lubbock schools. As such, Mary set up a lab in her elementary school and worked with all grade levels, concentrating very heavily on Logo.

A master's degree in instructional technology at Texas Tech gave Mary the opportunity to participate in an innovative program in Lubbock. The school district hired her as one of two programmers to create software. They found that just about the time they created a particular item, the commercial software would become available on the market, so Mary has become a software evaluator instead. In addition, she has been correlating the Texas essential elements (curriculum goals) with software, and making a catalogue for teachers.

Designs

Mary Upton's career centers on helping teachers use computers effectively to enhance learning. She has developed two guides for computer use with different curriculum areas. In these guides she has accumulated various Logo procedures that have been developed and has put them on a menu-driven disk. The guides also include a laminated turtle compass and teaching tips.

Plans for the near future include working extensively with Lubbock's 50 computer technologists to develop a support network, helping teachers use computers more as a tool, and fostering the burgeoning use of LogoWriter in the school district. The teachers who have started using LogoWriter are really excited and willing to learn. As Mary says, "There is just something special about Logo people!"

If you are interested in learning more about the computer technologist program or the materials Mary has developed, you may write:

Mary Upton
5420 15th Street
Lubbock, TX 79416

To apply for the UVA Logo Fellowship Program for the summer of 1988, request application forms from:

University of Virginia Summer
Logo Fellowship Program
Curry School of Education
Ruffner Hall
Charlottesville, VA 22901

If you know of someone who should be featured in this column, please contact:

Rebecca Poplin
2421 Fain Street
Wichita Falls, TX 76308

Rebecca Poplin uses Logo to teach computing and mathematics at a junior high school in Wichita Falls, TX.

Listful Thinking

by Glen L. Bull
and Gina L. Bull

Writing by Formula

Two summers ago we did a workshop at Punahou School in Hawaii. One of the products of that workshop was a Hawaiian syllable generator. A Hawaiian syllable generator relies on the fact that the Hawaiian language has only a very few consonants.

```
TO C
OP PICK [P L K H M N W]
END

TO PICK :LIST
OP ITEM 1 + (RANDOM COUNT :LIST)
:LIST
END
```

The procedure "C" shown above will pick one of those Hawaiian consonants. Since the consonants are picked randomly, the consonant you see when you try this procedure may be different than the one shown below.

```
?PRINT  C
M
```

The addition of a similar procedure to pick vowels, V, makes it possible to construct Hawaiian syllables.

```
TO V
OP PICK [A E I O U]
END
```

The consonants and vowels are combined in this way.

```
?PRINT ( WORD C V C V C V C V )
HAMAHANO
```

Of course, you could continue with as many C's and V's as you think necessary. This type of Logo procedure is simple, but makes a wonderful tool for study of language arts, phonetics, and linguistics. For example, what sorts of consonants would it be necessary to substitute in the consonant procedure to create a Polish syllable generator? Is it even possible to produce syllables that look Polish by only changing the consonants? What other factors could be involved?

This past summer, we introduced the Hawaiian syllable generator in the first week of a University of Virginia Logo class that had five science teachers and five elementary education and social studies teachers. The elementary education teachers thought the exercise was great fun, but the science teachers were more dubious. What, they wondered, did Hawaiian syllables have to do with teaching chemistry?

The standard explanation is that the syllable generator provides an introduction to list processing. The science teachers still looked unconvinced. At that point, a chemical compound generator was suggested. A compound generator, you see, could randomly pick chemical elements instead of syllables. It might look something like this.

```
TO EL
OP PICK [H Cu O F S Co Li]
END
```

It would be used in this way.

```
?PRINT ( WORD EL EL EL EL )
HCuLiS
```

When the chemistry teachers saw the result, they hooted. Chemical elements, it seems, can not be combined in any old way. Elements have valences, and the number of positive and negative valences in a compound have to equal one another. Here are some lists of elements sorted by valence.

```
TO PLUS1
OP PICK [H Li Na K]
END

TO MINUS1
OP PICK [F Cl Br I]
END

TO PLUS2
OP PICK [Mg Ca Ba Cu Zn]
END
```

```
TO MINUS2
OP PICK [S O]
END

TO PLUS3
OP PICK [Al Co]
END

TO MINUS3
OP PICK [N P As]
END
```

The positive valences and negative valences in a compound must balance. For example, in this instance an element with a positive valence of 3 is balanced by elements with negative valences of 1 and 2.

```
?PRINT ( WORD PLUS3 MINUS1
MINUS2 )
AlClO
```

In another case, an element with a valence of positive two is balanced by another element with a valence of negative two.

```
?PRINT ( WORD PLUS2 MINUS2 )
MgS
```

This contented the chemistry teachers, but not for long. They soon wanted to let Logo randomly generate the valences which would be used. For example, rather than the student deciding that the positive valence of Plus 3 would be balanced by elements with negative valences of Minus 1 and Minus 2, Logo could choose the balancing combination. This might be:

Minus 1 Minus 1 Minus 1 or
Minus 1 Minus 2 or
Minus 3

A random number between 1 and 3 can be generated with RANDOM.

```
?PRINT 1 + RANDOM 3
2
```

A student who wished to construct an element with a total positive valence of 5 might follow this train of thought:

"I need a total of five. OK. First I'll pick an element with a charge of 2. 5 minus 2 is 3. So I have 3 more to go. I'll pick a charge of 1. 3 minus 1 is 2. Then I'll pick a charge of 2. 2 minus 2 is 0. So I picked 2, 1, and 2."

The following Logo procedure works in much the same way.

```
TO GENERATE.CHARGES :TOTAL :LIST
IF :TOTAL = 0 [OP :LIST]
TEST :TOTAL > 3
IFTRUE [MAKE "CHARGE 1 + RANDOM
3]
IFFALSE [MAKE "CHARGE 1 + RANDOM
:TOTAL]
OP GENERATE.CHARGES ( :TOTAL -
:CHARGE ) ( LPUT :CHARGE :LIST )
END
```

To use the procedure, give it a total valence and the empty list [ ] :

```
?PRINT  5 [ ]
3  2
```

This gives us a list of positive valences. A list of negative valences is also needed.

```
?PRINT  5 [ ]
2  1  2
```

The two lists of randomly-selected valences provide all the information needed to create a compound. (Note that some of the compounds generated will "make sense" chemically, while many will not. What rules help you tell the difference?)

```
?PRINT ( WORD PLUS3 PLUS2 MINUS2
MINUS1 MINUS2 )
BaCoOFS
```

Now Logo can generate lists of positive and negative valences which can be used to write command to generate a compound. Could the computer automate the entire process? The heart of a program

to do this is a procedure which takes a sign (PLUS or MINUS) and a valence (1 through 3) and randomly chooses an element in that category.

```
TO RANDOM.ELEMENT :SIGN :VALENCE
OP RUN ( LIST WORD :SIGN :
VALENCE )
END
```

The procedure works like this.

```
?PRINT RANDOM.ELEMENT "PLUS 2
Cu
```

Another procedure uses RANDOM.ELEMENT to generate a list of elements.

```
TO GENERATE.ELEMENTS :SIGN :
VALENCES
IF EMPTYP :VALENCES [OP [ ]]
OP SE (RANDOM.ELEMENT :SIGN
FIRST :VALENCES)
          (GENERATE.ELEMENTS
:SIGN BF :VALENCES)
END
```

This procedure generates a random element for each valence in a list of valences.

```
?PRINT GENERATE.ELEMENTS "MINUS
[2 1 2] [ ]
SBrO
```

Sometimes the elements come out in jumbled order.

```
?PRINT GENERATE.ELEMENTS "PLUS
[1 3 1 1] [ ]
HCoLiH
```

A sort procedure could be used to sort the elements so that they come out in alphabetical order. A number of sort procedures have been developed in Logo. This one was created by Dan Weston and can be found at the end of the column. It sorts lists of words into alphabetical order.

```
?PRINT WORDSORT [Dog Bird Cat]
Bird Cat Dog
```

All of the procedures created thus far can be combined to make separate procedures to generate lists of elements with positive valences and lists of elements with negative valences.

```
TO POSITIVE.ELEMENTS :CHARGE
OP WORDSORT GENERATE.ELEMENTS
"PLUS
          ( GENERATE.CHARGES
:CHARGE [ ] )
END

TO NEGATIVE.ELEMENTS :CHARGE
OP WORDSORT GENERATE.ELEMENTS
"MINUS
          ( GENERATE.CHARGES
:CHARGE [ ] )
END
```

The procedures to generate lists of positive and negative elements can be combined in a superprocedure that makes compounds.

```
TO MAKE.COMPOUND :CHARGE
OP SE POSITIVE.ELEMENTS :CHARGE
NEGATIVE.ELEMENTS :CHARGE
END
```

The MAKE.COMPOUND procedure generates a list of positive elements equal to the charge specified and a list of negative elements with the charge specified.

```
?PRINT MAKE.COMPOUND 4
HMgFO
```

Just like the Hawaiian random word generator, some of the chemical compounds generated through this program would be valid, while most would not be. The point is that the compound generator provides a starting point for thinking about chemical patterns.

The combination of teachers and Logo produces teaching tools.

TEACHERS + LOGO = TEACHING TOOLS

Our thanks to all the folks at Punahou school in Hawaii for stimulating the syllable generator. Our

thanks to Esther Freeman, Joyce Corriere, and Cecelia Freeman for help with the chemical equivalent. Any inaccuracies in scientific content are ours, however.

A Logo tool is a procedure which you can incorporate in one of your own programs. Each month we hope to highlight a book which contains useful teaching tools. This month we would like to draw your attention to *The Second Logo Book* by Dan Weston (Scott, Foresman and Co., 1985). The following procedure for sorting words is from that text.

```
TO WORDSORT :LIST
IF EMPTYP :LIST [OP [ ]]
OP INSERT.WORD (FIRST :LIST)
(WORDSORT BF :LIST)
END

TO INSERT.WORD :WORD :LIST
IF EMPTYP :LIST [OP (LIST:WORD)]
IF LESS.THAN :WORD (FIRST :LIST)
[OP FPUT :WORD :LIST]
          [OP FPUT (FIRST :LIST)
(INSERT.WORD :WORD BF :LIST)]
END

TO LESS.THAN :WORD1 :WORD2
IF EMPTYP :WORD1 [OP "TRUE]
IF EMPTYP :WORD2 [OP "FALSE]
IF ASCII FIRST :WORD1 < ASCII
FIRST :WORD2 [OP "TRUE]
IF ASCII FIRST :WORD1 > ASCII
FIRST :WORD2 [OP "FALSE]
OP LESS.THAN BF :WORD1 BF :WORD2
END
```

Glen Bull is a professor in the University of Virginia's Curry School of Education, and teaches Logo courses at both the graduate and undergraduate level. His CompuServe number is 72477,1637. Gina Bull combines an undergraduate degree in History of Art with a graduate degree in Computer Science. She is a programmer analyst for the University of Virginia Department of Computer Science. By day she works in a Unix environment; by night, in a Logo environment.

Logo LinX

by Judi Harris

Blank Endorsement

Let's start the school year with the  's.

Do you recognize this?

It has 26 members; 6 of one type and 20 of another. It is also sung to the tune of a "pre-school golden oldie."

AHA! "Now you know your abc's...."

Mine's a Blank

This alphabet may have been difficult to recognize because it was written with BLANKWRITER, an unusual Logo letter-writer program that can help students with phonics, syllabication, and spelling.

I was inspired to write BLANKWRITER after working with Mrs. Rosalie Saul, a learning disabilities specialist from Danbury, Connecticut. Mrs. Saul created a teaching tool called "Alphablanks"; a modified wooden pegboard onto which students place colored plastic pieces representing letters of the alphabet. After seeing the power of the tool in action with Mrs. Saul's students, I enthusiastically borrowed her ideas and "computerized" "Alphablanks" in Logo.

Let's first examine the patterns in the blank alphabet to decipher the letter coding system.

You probably have already realized that the shapes of the BLANKWRITER letters are based upon a lowercase alphabet.

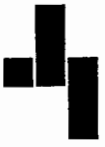
 "Up Letters," or b, d, f, h, k, l, and t

 "Down Letters," or g, j, p, and q


"Middle Letters," or a, c, e, i, m, n, o, r, s, u, v, w, x, z

Fill in the Blanks

Notice that some "middle letters" are represented by solid color squares, and others by square outlines. Look at the alphabet to determine the difference.

Consonants are depicted as blue solid-color rectangles 

Vowels are depicted as white outlined rectangles. 

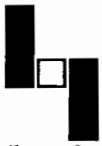
But, of course, to every good rule there is always an exception. The letter "y" sometimes behaves like a vowel, and at other times, like a consonant. It, therefore, is "blanked" as a solid WHITE rectangle 

@#&%* Blankety Blank Rules!

It's fascinating how easy it is to read words spelled in the blank alphabet. This, for example, is a word near and dear to our hearts...

...that looks much the same in blanks as it does in letters.

Some blanked words could be spelled several ways in letters,



(bag, beg, big, bug, dig, dip, dog, dug, fig, fog, hag, hip, hog, hop, hug, etc.) while others can only be spelled in one way.

(up)

(go)

Five "vowel rules" help children to see phonetic patterns in spelling, and therefore assist in decoding words.

1. bō  An open syllable ends with a vowel, which is usually long.

2. ŏb  A closed syllable ends with a consonant, and the vowel in the syllable is short.

3. rŏb  Another type of closed syllable begins and ends with consonants.

4. bōX  "When two vowels go walking, the first does the talking, and it always says its name."

5. rō bX  When a syllable contains this pattern, and the second vowel is a silent "e," the first vowel is long.

Students can learn to recognize these patterns by spelling out words that they know how to read in the blank alphabet. Then, when they are ready to tackle new words, they can use these syllabication rules before the vowel rules.

Pattern 1: 
Divide between consonants.

Pattern 2: 

If there is one consonant between two vowels, put the vowel with the first consonant if it is long.

Pattern 3:

 Give the consonant to the LE syllable.

Drawing a Blank

The procedures for making the BLANKWRITER are simple to code in Logo. Notice that each uses two tool procedures: RECTANGLE and OVER.

```
TO RECTANGLE :LONG :SHORT
REPEAT 2 [FD :LONG RT 90 FD :SHORT RT 90]
END

TO OVER :AMT
SETH 0
RT 90
PU
FD :AMT
PD
LT 90
END
```

RECTANGLE takes two inputs (the two side lengths,) and draws a rectangle of any size. Note that a square is coded as a special case of a rectangle in these procedures; one with the two side lengths of equal size. OVER takes one input, and automatically moves the turtle over after drawing each blank letter, so that it is in position to draw the next one.

The different types of blank letters are drawn with these procedures:

```
TO UP.LETTER
SETPC 1     <--Sets the pencolor to blue
SETH 0
RECTANGLE 30 10
SOLID.COLOR
OVER 12
END
```

```
TO DOWN.LETTER
SETPC 1     <--Sets the pencolor to blue
SETH 0
PU
BK 20
PD
RECTANGLE 30 10
SOLID.COLOR
PU
FD 20
PD
OVER 12
END

TO MIDDLE.LETTER
SETPC 1     <--Sets the pencolor to blue
SETH 0
RECTANGLE 10 10
SOLID.COLOR
OVER 12
END

TO VOWEL
SETPC 3     <--Sets the pencolor to white
SETH 0
RECTANGLE 10 10
OVER 12
END

TO DOWN.VOWEL
SETPC 3     <--Sets the pencolor to white
SETH 0
PU
BK 20
PD
RECTANGLE 30 10
SOLID.COLOR
PU
FD 20
PD
OVER 12
END
```

Consonants are filled in with the SOLID.COLOR tool:

```
TO SOLID.COLOR
PU
RT 45
FD 5
PD
FILL
PU
BK 5
LT 45
PD
END
```

Note: The above procedures were written in IBM Logo, but can easily be adapted to any version of Logo which includes a FILL command.

Blanking Out

I would like to extend a "point blank" offer. I have combined the procedures listed here with others to make a "blankwriter," or Logo-based letter coder that draws letters in the blank alphabet as the user types letters in the customary alphabet. The program allows up to three lines (of any length) of blankwriter words on the screen, and prints their translations at the bottom in lower-case letters. It also has built-in functions for spaces and erasing mistyped letters.

You may download this program from CompuServe's LogoForum (type go logoforum at any ! prompt); it is in Data Library 13. Or, you may send me a self-addressed, stamped envelope, and I'd be glad to send you the procedures in printed form. Finally, if you use IBM Logo or Logo II on the Apple, you may also send me a disk in a mailer with return postage, and I will put BLANKWRITER on it for you to use.

Judi Harris
University of Virginia
Curry School of Education
Department of Educational Studies
Charlottesville, VA 22903

Now, please tell me:

Judi Harris was an elementary school computer use facilitator, graduate education instructor, and computer consultant for a number of public and private schools in Pennsylvania. She is now a doctoral student in education at the University of Virginia. Her CompuServe number is 75116,1207.

Printing Long Lines in Apple Logo II

If you print the contents of your workspace from Apple Logo II or Logo II and you have lines longer than 80 columns, the lines will overprint and be virtually unreadable. The following procedures, adapted from a program provided by Alain Tougas of Logo Computer Systems, will allow you to print your procedures so that they are readable.

Note that you cannot print your procedures directly. You must first save your workspace to a file. Thus, you might type

```
SAVE "MY.WORK
PRETTY.PRINT "MY.WORK
```

to print the procedures which you currently have in memory.

```
TO PRETTY.PRINT :FILE
OPEN 1
SETWRITE 1
OPEN :FILE
SETREAD :FILE
ONE.LINE READWORD 78
SETWRITE []
CLOSE 1
CLOSE :FILE
END

TO ONE.LINE :LINE :LENGTH
IF EQUALP :LINE [] [STOP]
IF ( COUNT :LINE ) > :LENGTH
     [PRINT.LIMITED :LINE :LENGTH]
     [(TYPE :LINE CHAR 13 )]
ONE.LINE READWORD :LENGTH
END

TO PRINT.LIMITED :CHARS :LENGTH
LOCAL "REST
PRINT.LINE :CHARS :LENGTH 1
MAKE "REST REMOVE.FIRST :CHARS
:LENGTH 1
TEST ( COUNT :REST ) > :LENGTH
IFTRUE [PRINT.LIMITED :REST :LENGTH]
IFFALSE [PRINT.SHORT :REST ( COUNT
:REST ) 0]
END

TO PRINT.LINE :CHARS :LENGTH :COUNT
IF EQUALP :LENGTH :COUNT [( TYPE "!
CHAR 13 ) STOP]
TYPE FIRST :CHARS
PRINT.LINE BUTFIRST :CHARS :LENGTH (
:COUNT + 1 )
END

TO REMOVE.FIRST :CHARS :LENGTH :COUNT
IF EQUALP :LENGTH :COUNT [OUTPUT
:CHARS]
OUTPUT REMOVE.FIRST BUTFIRST :CHARS
:LENGTH :COUNT + 1
END

TO PRINT.SHORT :REST :SHORT.LENGTH
:COUNT
IF EQUALP :SHORT.LENGTH :COUNT [TYPE
CHAR 13 STOP]
TYPE FIRST :REST
PRINT.SHORT BUTFIRST :REST
:SHORT.LENGTH ( :COUNT + 1 )
END
```

A Problem Solving Contest... in Logo

Even though the school year is just beginning, you might want to circle April 30th, 1988 right now. On this date, the International Computer Problem Solving Contest will be offered. This contest, which has been run for the past seven years, first offered problems specific to Logo last year. These problems, aimed at elementary students, were well enough received that this year the Logo division will again be offered, but this time for all age levels. For more information about the ICPSC, write to Don Piele, University of Wisconsin-Parkside, ICPSC, Wood Road. - Bos #2000, Kenosha, Wisconsin 53141-2000 or call 414-5532231.

LXionary
A Lectionary of Selected Logo Readings

with Commentary and Opinion
by Bill Craig

A computer science teacher from a neighboring school division recently asked me at what grade my system "phases out" Logo. I explained that we were attempting to expand the use of Logo and unsuccessfully tried to convince him of the merits of those efforts. My inability to make a convincing argument forced me to review my Logo literature for the points that others have so eloquently made in support of Logo. I found my answers in something old and something new from Brian Harvey.

>> "Why Logo?" by Brian Harvey, *Byte,* August, 1982.

I have been guilty of claiming that Logo is a "powerful" language without really knowing what the claim means. Harvey makes a powerful arguement for the power of Logo as a computing language and as an educational tool. The first part of the article details the strengths of Logo compared with other computer languages such as FORTRAN, LISP, and BASIC. Included are discussions of what a procedural language is and why it is important, comparison of compiled versus interpreted languages, means of treating variables, and methods of list processing. This section provides a great primer in computer science for a non-computer scientist like me. Harvey also uses the article to support the use of Logo as an educational tool. Many of the arguements are familiar (no threshold, no ceiling), but the message is timely.

All of us have had to answer the question that serves as the title to this article. Harvey provides us with the ammunition to answer that question with authority.

>>"Finding the Best Logo for Your Students." by Brian Harvey, *Classroom Computer Learning,* April, 1987.>>

Once you have convinced yourself and others of the value of Logo, a version of Logo must be selected. An indication of the growth of the Logo movement is a chart in this article which compares fourteen versions of Logo for six different computers. Harvey's article discusses hardware considerations, size of memory, speed of operation, bugs and "misfeatures," and a comparison of MIT and LCSI versions of Logo. The accompanying chart lists characteristics such as price, memory requirements, speed, and method of debugging for fourteen Logo versions.

Not surprisingly, Harvey does not endorse one implementation of Logo as the "best." He makes the very safe statement that the version you choose will be determined by what projects you attempt and the hardware that you have. He closes by saying that, "Logo is not a finished product; look forward to new ideas that will extend the language in various directions."

Bill Craig is the Assistant Principal, Hening Elementary School, 5230 Chicora Drive, Richmond, VA 23234.

PenPoints

Logo Book Reviews with ASTROLUG

"You, the teacher, are the single most important factor in achieving an effective implementation of Logo. The tone and nature of the class are just as important as any materials used in the class."

Peter: Hi, and welcome to PenPoints, a column written by ASTROLUG members to review Logo books. ASTROLUG is the Albany - Schenectady - Troy Logo Users' Group in upstate New York. Across the keyboard from me is Craig Hodgson, a K-5 Computer Resource Teacher. Craig teaches over 1100 elementary students in three different buildings. He works with them in their classrooms once a month.

Craig: Across from me is Peter Rawitsch, a first grade teacher who teaches Logo courses and workshops to area educators. Our opening quotation is from a new book, *The LOGO Success Kit*, by David Chesebrough. The "kit" is a 359-page Logo curriculum that is full of reproducible classroom materials for middle school students. It is divided into eight Logo skill levels of increasing difficulty. (Level 8 explores word and list operations.) Instructions for Terrapin Logo, Apple Logo and Apple Logo II are provided. A Resource Disk of procedures, pictures and programs is also included. It is published by J. Weston Walch (321 Valley Street, PO Box 658, Portland, ME 04104-0658, telephone (800) 341-6094).

Thorough Logo Kit

C: I am excited about teaching Logo to young children. However, I feel as though I've had to reinvent the wheel many times as I've introduced Logo concepts to my students. I found lots of ideas in *The LOGO Success Kit* that I will use next year and I'm sure that teachers who are new to Logo will find it has a lot to offer them as well.

P: What impressed me the most was how complete it is. It doesn't pretend to have everything and yet you can go through all of the different sections and find that almost every issue that needs to be addressed is covered. The author discusses concrete learning, teacher preparation, and microworlds. I might not agree with him about all of the issues, but at least they are addressed.

C: I like the whole section on sequencing a program. Four-week, ten-week, 18-week, full-year and three-year programs are outlined. That's a lot of options for school systems. People who are starting with Logo may not have any idea how long it would take to get through a sequence of activities.

P: He also mentions that when students begin working on projects, it's possible that it will take a lot longer than it seems on paper. If you only have one computer in your classroom, it makes a big difference as to how quickly your kids can complete a project. Allowing extra time for projects is suggested.

C: As a computer teacher, I can see this book as a valuable reference. The glossary is very handy for looking up something I want to check. The book also provides different ways of presenting material to students or teachers who have difficulty underst anding some of the harder Logo concepts. His bibliography is very helpful for a person in my position who is trying to build up a professional library. Many teachers ask me for books with ideas to supplement Logo activities. I found the Troubleshooting Guides interesting. He's described some of the problems that my teachers have actually experienced. If I gave them a copy of one of the Troubleshooting Guides, it would save time.

Flexible Activities and Projects

P: One of Mr. Chesebrough's rationales for teaching Logo is, "to provide an environment where students are in control of their learning and the computer." In going through these materials I thought to myself: "Are these activities that empower students?" My initial reaction was no, they don't, because they're very teacher directed. On the other hand, he provides for open-ended projects. Many times he states that his activities are just to get the kids started and if they have other ideas, that's okay. I love this quotation from the book:

"If a fascinating challenge or project idea develops in the class or with a group of students,

throw any sequence plans out the window and let the students explore and enjoy, supporting them with material as needed from *The LOGO Success Kit.*"

C: When I started teaching Logo I was really afraid of using worksheets. I thought Logo should be more of an exploratory activity. This year, I found I could use them, because I set up the rules for how the worksheets were to be used. The book has good activity sheets. I can give them to teachers as a starter or model activity. In the wrong hands, of course, they could be abused. It's called a "Logo Success Kit," but as Chesebrough says in this book, it is the teacher who will make it successful.

P: Even though the book is designed for use with middle school students, I felt the layout of the activity sheets was too cluttered. They would be more readable if there was less text.

C: They do look like the typical writing or reading exercises found in workbooks that kids see everyday. Some of them will say, "Oh, no. Not another one of these." What I like are the activites themselves. For example, the Beginner's Section (Levels 1 - 3) was organized well and paced appropriately.

P: Many times kids are taught to memorize the syntax rules without any real understanding. In the Advancing User's Section (Levels 4 - 8), a better explanation should be provided as to why a quote is used for the name of a file or a colon for variable names. Middle school kids could handle the idea of how Logo evaluates procedures and input.

C: The Resource Disk is really helpful. It offers "visual" explanations for many of the projects. I can show a class a sample quilt as a model of the finished product. Everything I've demonstrated to my students this year I've had to create.

Skill Checks vs. Tests

C: Parents and administrators are interested in seeing what the kids are learning on the computer. The Review / Skill Check at the end of each level will help teachers show what has been learned. I know it doesn't really go with the Logo philosophy, but he says specifically these are not tests.

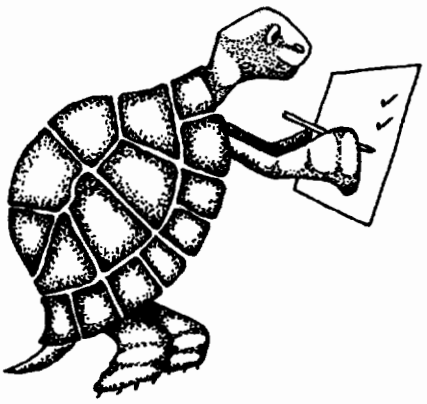
P: I think it's silly not to say these are tests. They are tests of skills. If a child is unable to demonstrate his or her ability to do a skill, you go back and reteach that skill. It struck me that he was being a little overprotective of Logo. If you're trying to teach a specific skill, such as saving a picture, it's okay to test a child to see whether or not they're able to do it. If you have children who are new to your school, a Review / Skill Check can be used as an assessment tool to determine which Logo language skills they already have. A more meaningful "test" for me would be one that determines to what extent the children were "in control of their learning."

Two Penups!!

P: Now, let's recap our reactions to the The LOGO Success Kit. I give the book a "penup". (An ASTROLUG "penup" is equivalent to a "thumbs up.") I like its thoroughness and flexibilty.

C: The fact that a book like this is now available excites me. Regrettably, I'm only able to see students once a month, which forces teachers to work with their students alone. Not all of them want to put their kids on CAI disks when they're done with my follow-up Logo assignments; they want to go further. I can't always get back to them. Now I can leave them with a whole collection of good materials. Make that two "penups."

P: ASTROLUG will be back next month with a review of another Logo book. Until then, keep reading!



ObjectLogo: A Review

by Robs Muir

As Andy diSessa tells it, one of the early criticisms of the Logo language was "it just won't work." According to the critics (circa 1970), Logo's emphasis on graphical displays was too memory-intensive for any realistic implementation on an affordable educational computer. Yet, as we all know, the Logo turtle survived intact. Much of the power of the original mainframe Logo was successfully transferred to our modern personal computers. What many of us wonder is, what was lost in the translation?

Logo does tax the computational resources of an eight-bit microprocessor. By the late-seventies, as microcomputer-based Logos were being shoe-horned into 64K of RAM or less, significant compromises had to be made to bring Logo to the masses. Many of these pragmatic decisions were carefully made so as to not compromise the integrity of the Logo philosophy, but the language was pared down none the less.

Logo was intended as a language with "no threshold and no ceiling"—easy for children, yet powerful enough for sophisticated computer scientists. Logo for most microcomputers handles the no threshold part quite well; many of us have been waiting for the ceiling to be raised. Logo's ceiling is a technological reality imposed on our imagination.

For those of us who started using Logo on microcomputers, many of Logo's limitations became increasingly apparent as we learned more about our new language and our own burgeoning programming skills. As we worked with early versions of Logo, the "low threshold" enticed us to introduce our youngest students to the pleasures of creative interaction with a machine. Yet as we became more proficient, both students and teachers noticed an unannounced "low ceiling." Usually the ceiling was a lack of node space—the OUT OF SPACE error.

Logo fell just short of its promise. However, it was more fun than the alternatives. And until a better version of Logo came along, programmers still had lots to play with. We had a full-screen editor, recursion, list processing, local variables, and no line numbers! What we lacked was speed, space, expanded data structures (arrays, strings, etc.), better input / output, and low-level access to the machine. In short, we wanted POWER! We wanted Lisp without the parentheses. We wanted APL without special keyboards. We wanted a serious implementation that would allow us to teach AP Computer Science without dying of "cancer of the semicolon." All of this on under $1500 worth of hardware.

Consider the following text a description, rather than a review, of a new and exciting implementation of Logo. This "industrial strength" Logo is simply too big and too new for any complete assessment by any one individual.

Raising the Standards

The newly released ObjectLogo from Coral Software holds great promise for the Logo programmer. Designed for the Apple Macintosh (512, 512E, Plus, or SE), this new implementation offers new features, unique possibilities, and significant power. The remarkable feature of ObjectLogo is that it provides a rich new programming environment while maintaining a strong link to Logo's traditions.

ObjectLogo is an outgrowth of QLogo, an object-oriented version of Logo developed at the Atari Cambridge Research Center which closed in 1984. Coral Software was founded to bring to market both a full-featured Lisp and a powerful Logo to the newly announced Macintosh. Coral needed over two years, first, to build a Lisp compiler, and then to write ObjectLogo within this custom Lisp. As such, ObjectLogo closely shares its Lisp lineage; little computational power seems to have been lost in the translation from Lisp to ObjectLogo.

Major Features

ObjectLogo is an object-oriented language built around a paradigm defined by Gary Drescher (an early Logo worker). Object-oriented programming is a style of programming currently in vogue among computer scientists which promotes modularity by providing a mechanism for building related computational structures which may share data. While supporting this new

programming mechanism, ObjectLogo offers nearly complete compliance with older Logos, specifically LCSI's Apple Logo. This makes it possible to use ObjectLogo nearly as easily as traditional Logos; one is not forced to use object-oriented programming.

Object-oriented programming is ideally suited for multiple turtles, since each turtle must keep track of its own position, heading, penstate, etc. ObjectLogo does permit an unlimited number of turtles; additionally, each turtle may "own" its own version of a procedure. FORWARD 100 may mean different things to different kinds of turtles. ObjectLogo permits different types of objects other than turtles to exist, so it is possible to create multiple text and graphics windows, editors, and listeners. ObjectLogo includes an advanced input/output (I/O) scheme using streams which allows generalized I/O for random-access files, windows, and serial I/O. These things are all possible because of object-oriented programming. (See Jim McCauley's "Q and A" column in April 1987, Vol. 5, Number 8, *Logo Exchange* for more information on object-oriented Logo programming.)

While offering words and lists like any reasonable Logo, ObjectLogo also permits arrays, objects, and MacTypes (a data type specific to the Macintosh environment). ObjectLogo provides for complete access to *all* of the QuickDraw routines (drawing tools) resident in the Macintosh ROM, providing additional speed and control for graphics programs. For low-level programming, ObjectLogo also gives complete access to Macintosh environment, with .ADDRESS, .GET, and .PUT, and access to the Mac's Stack and Trap.

ObjectLogo gives Logo the power of many modern Lisps. A programmer may build circular lists using equivalents of Lisp's REPLACA and REPLACD. ObjectLogo includes a .JOIN which functions like a CONS in Lisp. While this will appeal only to those interested in computer science, ObjectLogo is the strongest indication yet that Logo is a mature programming language.

A departure from traditional Logo is ObjectLogo's use of lexical scoping vs. dynamic scoping. This insures that the variables used within a procedure are strictly local to the procedure in question. In effect, this means that a procedure's variables are LOCAL unless otherwise stated. Dynamically scoped variables, while available, must be explicitly declared. This change in Logo reflects the changes within the Lisp community during recent years where dynamic scoping has also given way to lexically scoped variables. Again, lexical scoping promotes modular program design.

A down-side to this scoping issue is the incompatibility of some existing Logo programs with ObjectLogo. If the original programmer used a programming style that relied heavily on dynamic scoping, some programs may require some rewriting for ObjectLogo. While much of ObjectLogo is syntactically similar to LCSI's Logos, there are some differences. For example, PPROP has been changed to PUTPROP. I am also sad to report that WRAP and FENCE are not in the primitive set. The commands for packaging are not implemented in ObjectLogo, though objects achieve much of this functionality.

ObjectLogo supports over 700 primitives, and as such is an intimidating implementation. These new primitives help provide a mathematics package as complete as Common Lisp's. There are several data types which are new to Logo. Included are "unbounded integers" (extremely large integers are accurately represented internally and do not thereby lose precision), floating point numbers (.5 + .25 = 0.75), rational numbers (1/2 + 1/4 = 3/4, not 0.75), and complex numbers (now you can find out the square root of negative numbers!). ObjectLogo supports the widest range of transcendental and hyperbolic functions that I have seen in any Logo.

Most Logos give us the wrong number when we use large integers. The expression,

```
?SHOW (PRODUCT (QUOTIENT
112233445566778899 2) 2)
```

usually returns the value of 1.12233E17. We have lost some precision. This is not at all the same as 112233445566778899 which ObjectLogo returns. For mathematicians, ObjectLogo is a step in the right direction.

New to Logo is ObjectLogo's implementation of optional inputs to a procedure. In traditional Logos, we are used to using parentheses to include additional inputs to

a command or operation, i.e., (SUM 1 4 3 2 5). ObjectLogo allows programmers to write procedures that also provide for optional inputs. For example,

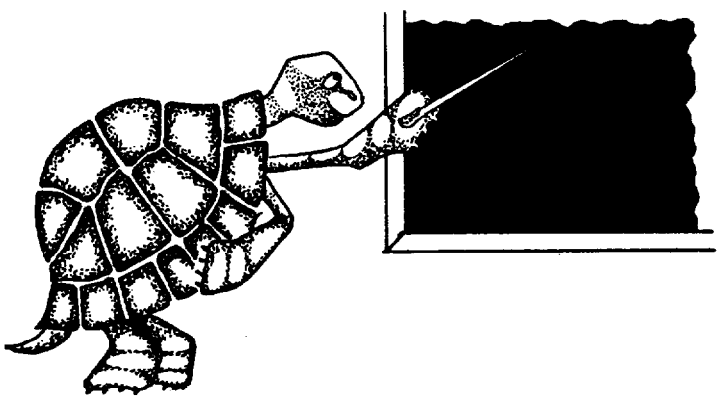
```
TO AVERAGE [:NUMS]
OP (APPLY "SUM :NUMS) / (COUNT :NUMS)
END
```

will allow for (AVERAGE 34 45 12 18 6). ObjectLogo also provides a mechanism to specify the default number of inputs, e.g., the primitive LIST normally requires two inputs. Its default is two. More or less inputs than the default must be delimited by parentheses. Coupled with optional inputs, this allows ObjectLogo programmers to build procedures with similar behaviors as the various primitives.

ObjectLogo is the fastest Logo that I have yet tested. This is, in part, because it is incrementally compiled into machine language during execution. This is a sharp contrast to most available Logos which are interpreted. ObjectLogo's compilation process is transparent to the user—it just seems to run faster than interpreted Logos.

To get an idea of the relative speed of ObjectLogo compared to other Logos, I wrote a simple turtle graphics program. Here is a spiral benchmark program and the resulting time with the turtle visible (ST) and with the turtle hidden (HT)

```
TO SPIRAL :DIST :LIMIT
IF XCOR > :LIMIT [PRINT "DONE!! STOP]
     FD :DIST
RT 1
SPIRAL (0.001 + :DIST) :LIMIT
END
```



Benchmarks using SPIRAL 0.1 100 (after doing a RECYCLE/.GCOLL)

	ST	HT	POS
ObjectLogo	1:23	:27	[100.246 21.340576]
LCSI Logo (Mac)	1:41	1:29	[100.247 21.34073]
Apple Logo	3:06	2:28	[100.237 21.3351]
Apple Logo II	2:44	2:10	[100.237 21.3351]
Terrapin Logo	4:01	2:23	[100.237 21.3351]
Commodore Logo	3:20	2:50	[100.237 21.3351]
IBM Logo	3:08	1:35	[100.244 21.341]
LogoWriter (Apple)	1:50	1:35	[100.2471 21.3391]
ExperLogo (Mac)	(Failed to run; integer turtle graphics only.)		

ObjectLogo includes some excellent debugging tools—only to be expected in such a sophisticated, modern language. Also included are a variety of useful tool procedures, such as MAP, FOR, FOREACH, FILTER, WHILE, and WHEN. Additionally, the most recent version of ObjectLogo (v. 1.2) includes MIDI interfacing and the cutting and pasting of graphics images.

ObjectLogo has provided a comprehensive, professionally printed reference manual with a 24-page index! Though sparse in examples, this manual provides an adequate reference for the experienced Logophile.

Coral Software is offering this package for $79.95. Compared to other implementations of Logo, and given the power and complexity of this program, ObjectLogo would be a good value at twice the price.

Clearly, this is a rave review. Excuse me...a rave description. Even so, there are several things I'd recommend to improve the package.

ObjectLogo Wish List

Coral has promised a compiler that would allow ObjectLogo code to be fully compiled into 68000 machine code. This would permit the building of standalone applications written entirely in Logo! I'd like to see this delivered as promised.

ObjectLogo is extremely faithful to the spirit of Logo and, as such, it is relatively simple to move into this new programming paradigm. However, there are still a few rough edges. Error statements, while similar to those available with LCSI Logos, are sometimes unhelpful, for example, (my favorite) "SOME PRIMITIVE DIDN'T GET A VALUE." Also, it is still possible to "crash" the language—at least in this early release version.

I'd like to see the complete range of Macintosh dialog boxes, alert boxes, and selection boxes implemented as object types.

I do miss having the EMACS-like commands for keyboard editing (CTRL-A, CTRL-D, CTRL-E, etc.) Having to use the mouse is somewhat inconvenient for this Apple Logo trained programmer. Also, the arrow keys on the Mac Plus have not been implemented in the preliminary version I have been using (v. 1.1).

ObjectLogo really needs some supplemental materials to help explain the intricacies of object-oriented programming. Introductory tutorials are badly needed. I'd like a hardcover copy of the reference manual. I'd even pay extra for it. With a language this substantial, the manual really gets a workout. My copy is already dog-eared from overuse.

What I'd really like is much, much, much more free time to use this exciting new vision of what Logo could be. Perhaps Coral could include some release time from my classroom? A substitute, perhaps??

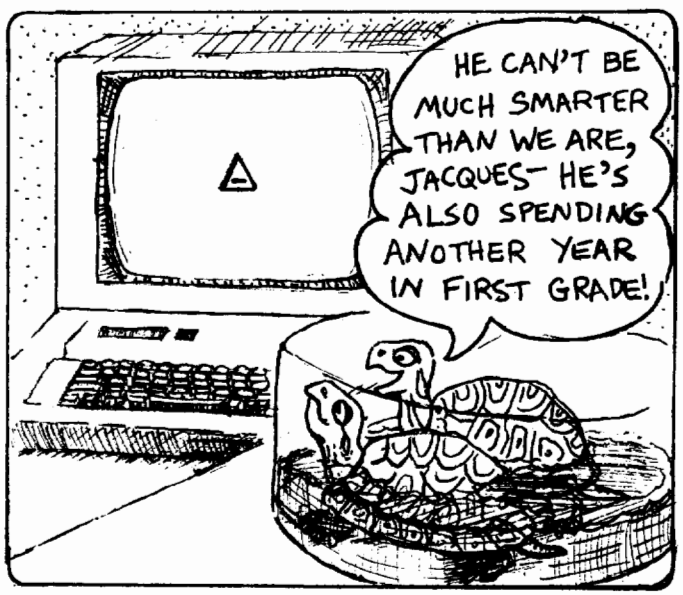
ObjectLogo is available from: Coral Software, PO Box 307, Cambridge, MA 02142, or call (800) 521-1027.

Robs Muir is a physics and computer science teacher in Claremont, CA, and an instructor at the Claremont Graduate School. His CompuServe number is 70357,3403, and his Bitnet address is MUIR@CLARGRAD.

Jacques and Elsie

drawings by Linda Sherman

Here are Jacques and Elsie ready to start the new school year. Next month we will again print a captionless Jacques and Elsie cartoon by Linda Sherman, and ask you for suggestions for the caption.



ROUTE 1, BOX 292-A
SHIPMAN, VA. 22971
(804) 263-4036

By way of reintroduction:

Jacques & Elsie, two familar, garden variety "classroom pet" turtles, first encountered a Logo turtle last fall. After a summer separation, they are delighted to meet again with their computerized counterpart!

Linda Sherman is a freelance author and artist living in Shipman, VA., with her husband and two-year old son.

IntLXual Challenge

by Robs Muir

Fractal Frolic

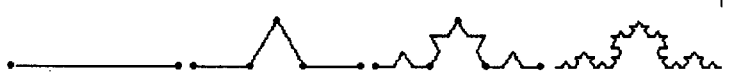
When I was first introduced to Logo (and its "cute" turtle), I was well aware that the turtle was considered to be a transitional object for the neophyte. This is why children are introduced to Logo through turtle graphics. Young learners need such experiences to create a bridge between concrete experience and abstract ideas; the turtle (according to Papert) was to be a concrete bridge leading into Mathland. Turtles are *body syntonic*.

However, I was learning Logo as an adult—not as a child. I still remember my impatience with drawing "pretty pictures." It was not until I received a copy of Abelson and diSessa's *Turtle Geometry: The Computer as a Medium for Exploring Mathematics* (MIT Press, 1980) that I realized that turtles were not merely for children. Here were microworlds of extraordinary depth.

I continue to be amazed at the richness and the (apparently) endless possibilities for exploration.

This month's challenge is an extension of a class of turtle problems that have been widely, yet superficially, covered in many Logo tutorials and Logo textbooks. I am referring to *fractals*. If you are a serious Logo learner, you have doubtless been exposed to examples of fractal programs as you learned about everyone's favorite topic, recursion. The term, *fractal*, was coined by Benoit Mandelbrot, an IBM Fellow and the self-proclaimed father of fractal geometry.

Below is a classic example of Logo fractals which has been presented in innumerable workshops and textbooks. While Mandelbrot calls it a Triadic Koch curve, it is called (variously) a *snowflake* by Abelson and diSessa, or *K3* by Thornburg (in Apple Logo: *An Invitation to the Art and Pattern of Nature* (Addison-Wesley, 1983)). Mandelbrot also calls it a snowflake *K*. In case you missed it, here it is yet again.



SIDE 50 0 SIDE 50 1 SIDE 50 2 SIDE 50 3

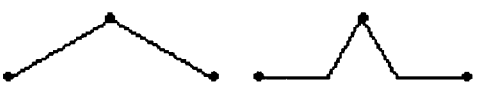
```
to side :size :level
if :side = 0 [fd :size stop]
side :size / 3 :level - 1
lt 60
side :size / 3 :level - 1
rt 120
side :size / 3 :level - 1
rt 60
side :size / 3 :level - 1
end
```

Mandelbrot describes this fractal using the following notation. The number of segments is represented by N and the scale factor of each successive reduction is represented by r. The letter D represents the "fractal dimension."

It is interesting to note that this same fractal

$$D = \log N / \log(1/r)$$

sequence can be generated in a slightly different way. Invented by Ernest Cesàro, this method relies on an inverted "v" for the first stage.



$$D = \log 2 / \log \sqrt{3} \approx \log 4 / \log 3$$

The SIDE procedure can be used to create a larger program that will draw a complete Triadic Koch Island as follows

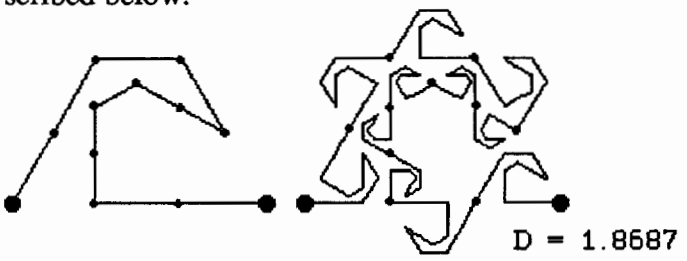
```
to triadic.koch.island :size
:level
repeat 3 [side :size :level rt
120]
end
```

Challenge #1

Can you recreate Cesàro's Koch Island in Logo?

Challenge #2

The following fractal, called a *Monkey Tree*, occurs on page 31 of Mandelbrot's *The Fractal Geometry of Nature* (W. H. Freeman and Company, 1983). It is an alternating fractal of a type which I have yet to see discussed in any Logo book. At first glance, I expected to recreate this in Logo easily since it is a recursive design based on an initiator as described below.



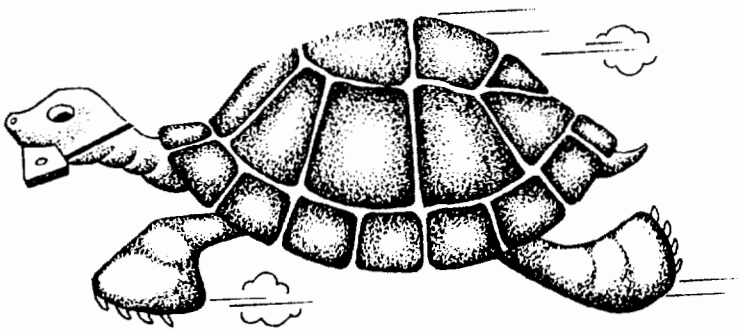
Unlike other problems presented in this column, as I write this I still have not solved the Monkey Tree for any "level." The complexity of this fractal is hidden by its similarity to other more familiar algorithms. This, then, is left as an exercise for the reader; may it serve you well.

Send listings of your efforts in a stamped self-addressed envelope to:

IntLXual Challenges
c/o Robs Muir
1688 Denver Avenue
Claremont, CA 91711

I will redistribute copies of interesting solutions (if any!) to all contributors.

Robs Muir is a teacher in the Claremont (CA) public school system and a member of the faculty of the Claremont Graduate School. His CompuServe number i s 70357,3403, and his Bitnet address is MUIR@CLARGRAD.



International Logo News

edited by
Dennis Harper

The use of Logo continues to grow by leaps and bounds throughout the world. The LX international network of continental editors and contacts is also very much alive and expanding. The International Logo News will now be found in one column with many contributors.

I have completed my stay at the Institute of Education in **Singapore** and am now a visiting professor of computer education at University of Helskinki in **Finland**. Part of my duties involve working with a Logo research team in the primary schools. More about this in later issues.

João Filipe Matos of the faculty of education at the University of Lisbon in **Portugal** reports that seventy teachers from throughout Portugal attended a three-day Logo Week. The event was organized by the Universities Portalegre School of Education and the MINERVA Project (aimed at introducing computers into the primary and secondary schools both as a subject and as a learning tool). This is the first time a large number of Portuguese teachers involved in Logo got together specifically to discuss and share their experiences and difficulties in using Logo.

Hands-on activities in workshops on music / Logo, LogoWriter, Sprites, Lists, and Portuguese Logo helped to create the appropriate environment to discuss Logo teacher training issues. Several papers reflected the experiences of the MINERVA Project. Logo is one of the main priorities in their work with teachers and students. Four schools of education in the country are working with Logo and a "Logo culture" is emerging among Portuguese teachers and teacher trainers. The enthusiasm about new technologies in education is now increasing, and it is possible that Logo can really get a major foothold into the Portuguese schools.

Patrick Scott tells of a different Logo experience in rural **Guatemala**. In July 1986 Guatemala's National Bilingual Education Program began such an experience with two Apple II+ computers loaned by the University of **New Mexico** along with a version of Spanish Logo. The experience began at a rural indigenous primary school ten miles from the nearest town and above 8000 feet in elevation. Luckily, a major electrical power line runs through the area.

A newly appointed teacher was given a very short training course in Logo and a copy of Reggini's *Wings of the Mind* (*Alas para la Mente*) and was sent to integrate Logo into the school's curriculum. The first graders who were the main Logo users arrived at the school speaking only the local Indian language.

The children took to the turtle as readily as their contemporaries from other cultures, and soon had it dancing across the screen. Some members of the community have expressed pride at being the only rural elementary school in Guatemala that offers its students a chance to have experience with a computer. Only two negative comments have been received. One from a local gentleman who complained that such technology was not indigenous. He had nothing more to say when it was pointed out that his clothing was not indigenous either. The other complaint was from a small group of teachers at surrounding schools who heard of the project and feared that the scarce resources of the bilingual program had been used to purchase the computers. They were assured the computers were free on loan.

A mathematics pretest was given to a Logo group of 26 first graders and a control group of 43 first graders. Although space limits details of the study, the following chart gives an indication of the Logo group's success.

	Pretest		Posttest		
	Mean	S.D.	n	Mean	S.D.
Logo Group	47.0%	19.8%	26	80.2%	13.8%
Control	50.1%	14.6%	43	42.9%	16.1%

The initial success of the project was sufficient to encourage those involved to try it for another year. The teacher is so motivated that rather than take the only bus that passes through the village to get back to his home right after school is out, he stays late to work with students and then walks over seven kilometers down to the next larger town to get a bus!

From the LX Asian Correspondent — Hillel Weintraub

Dr. Chris Templar of Johnson Bible College in **Tennessee** was recently in **Japan** and **China** discussing Logo with graduate students and professors. She delivered a talk at Beijing's Research Institute entitled "Integrating Logo into the School Curriculum."

Japan's Takito Nishimi is extremely active these days, now putting the finishing touches on his 1000 page Logo study which will be published in Japanese by the end of this year. We're hoping that an English version will be published as well.

Dr. Yamanishi has been working with Lego-Logo development in **Japan** and held a three-day workshop in August. As most workshops for teachers in Japan are usually one or two days, this longer training is a move in the right direction. Professor Yamanishi is also continuing his work with Logo and handicapped children and is now preparing the results of his research.

Helen Turner, the Elementary Computer Specialist at the International School in Bangkok **Thailand**, is concentrating on list processing with her 600 students. The children decide on their Logo goals, e.g., sentence generation, Haiku poems, and Madlibs. Ms. Turner has found many activities appropriate for elementary children using Logo list processing. She finds that the main advantage of using list processing over Logo graphics is that the list processing activities can be used in conjunction with the homeroom teacher's curriculum. It is found that graphics are not as easy to integrate into the curriculum and most homeroom teachers do not understand their educational value.

Tom Lough served as an educational computing specialist with the "Information USA" exhibition in the **USSR** recently. The exhibition was part of the new cultural exchange program and was sponsored by the US Information Agency. Tom showed Logo and LogoWriter to many teachers, and delivered a talk to the Lenin Pedagogical Institute in Moscow, the leading teacher training institute (10,000 teacher candidates in 15 disciplines). He reported that the teachers and teacher trainers became very excited about the potential of Logo, and hopes that the use of the language will spread as computers are introduced into the Soviet education system.

If you have Logo news of international interest, please send it to one of the continental editors below.

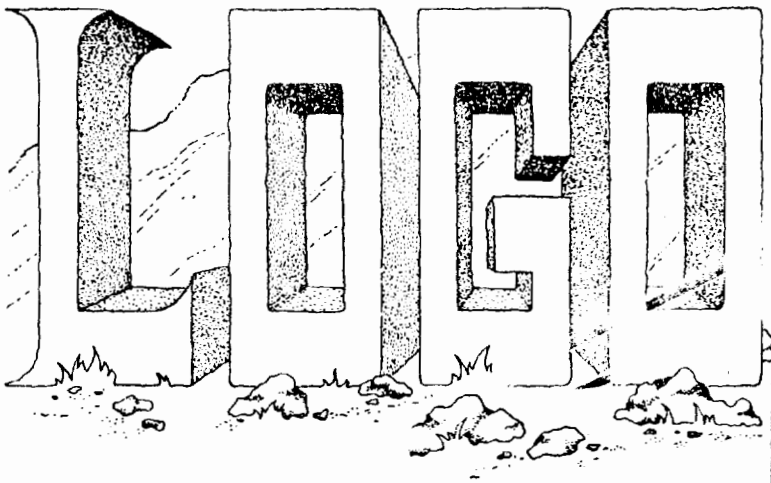
Africa
Fatimata Seye Sylla
Lab Informatique et Ed
BP 5036 Dakar
Senegal, West Africa

Asia
Hillel Weintraub
Doshisha Int. Sch.
Tatara, Tanabe-cho
Twuzuki-gun
Kyoto-fu, Japan
610-03

Australia
Anne McDougall
Faculty of Education
Monash University
Clayton, Victoria 3168
Australia

Europe
Richard Noss
London U. Logo Group
IE, Dept. of Math
20 Bedford Way
London, UK WC1HOAL1425
Argentina

Latin America
Eduardo Cavallo &
Patricia Dowling
Instituto Bayard
Salguero 2969
Buenos Aires



LogoPals

by Barbara Randolph

Logo Penpal Network

Congratulations, LogoPals! Celebrate with us the first anniversary of our international Logo student network! Exactly one year ago, the first official request for LogoPals was published here. Our first LogoPals and their ambassadors (their teachers) served the network well by being the pioneers of a new worldwide Logo adventure.

Consider this an open invitation to become involved in our LogoPals project (or to continue your participation). LogoPals is an international network of students who write to each other to share ideas about their work with Logo. And that's not the only fun! They also have the opportunity to get to know someone new and discover what these new friends enjoy doing, and find what it's like to live in another part of the world. LogoPals can both learn more about Logo and explore their planet through a new friendship.

Can you believe it? When I was young, there were no computers in my school, not even a calculator. And of course we had never even heard of Logo. But I was lucky enough to have penpals from cities and towns in Japan, Nigeria, the Philippines, the United States, and other places. I have wonderful memories of those friendships which came alive for me through letters and pictures and even one visit from a penpal from Hong Kong. Unforgettable!

What an adventure can begin for Logo students around the world! And, LogoPals, you'll be excited to hear that we plan to connect this year with teachers and students in China, France, Tasmania, Japan, Russia, and other ports of call. That new friend from that special place you request from us may be just around the corner!

How can Logo students become Logo penpals? They can each write to me, telling their age and grade, hobbies and interests, and their favorite Logo activities. They can be matched with any of the boys and girls in our LogoPal network. Students from primary through high school grades can participate. Skill levels vary from beginner to advanced. Requests for penpals from particular areas will be accommodated where possible.

Children in the USA need to send a self-addressed stamped envelope. Those outside the USA should enclose international postal coupons (purchased at the post office) for a 1-ounce or 28-gram reply.

A number of "ads" (excerpts from letters) will be printed in the LogoPals column each month. Teachers, please make sure that the *Logo Exchange* is available to your students because they love to see their ads featured on our LogoPals page!

We appreciate the enthusiasm about Logo and LogoPals that you generate with your students. Your help and encouragement keeps those letters and ideas flowing!

Don't delay. Write today. Write to:

Logo Pals
c/o Barbara Randolph
1455 East 56th Street
Chicago, Illinois 60637
USA

Logo Class Penpal Network

If you would like to involve your entire class in a Logo project, then join the Logo Class Penpal Network. Through the network, you and your class will establish a correspondence with another teacher and class in the United States or in another country. You will be able to share Logo teaching and learning ideas, procedures, and projects. To obtain a free application kit, send a long self-addressed stamped envelope to:

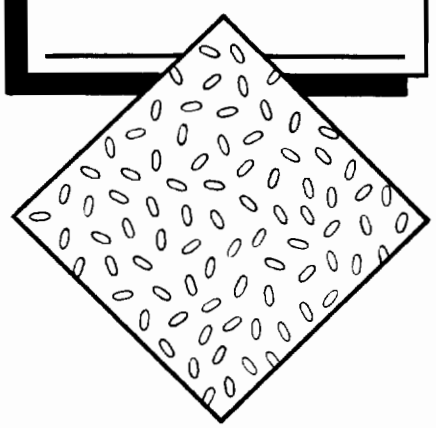
Logo Class Penpal Network
University of Virginia
Curry School of Education
Ruffner Hall
Charlottesville, VA 22903

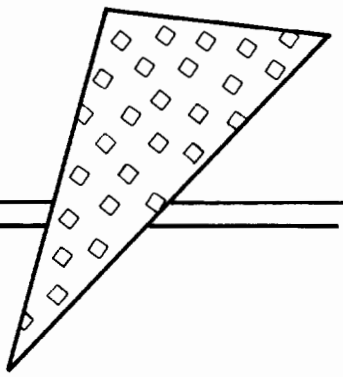
Barbara Randolph is a librarian and instructional media center teacher in the Chicago Public Schools.

Join the Leading Professional Organization for Computer Educators!

Whether you are an administrator, curriculum specialist, classroom teacher, media specialist or special educator, you will benefit from membership in ICCE.

ICCE is the leading U.S. and international professional organization for computer educators. It is non-profit, supported by more than 50 organizations of computer-using educators worldwide.





Membership in ICCE includes a subscription to *The Computing Teacher* journal. The journal has long been respected as an important source of information for computer educators, providing accurate, responsible and innovative information. Nine times per year, its articles, departments and reviews keep you in touch with the constant changes taking place in educational technology.

Members will also receive discounts on all ICCE publications and SIG memberships, and information on ICCE committee activities.

As educational technology continues to grow and change, look to ICCE and *The Computing Teacher* journal to keep you on the leading edge.

Join today!

ICCE, University of Oregon, 1787 Agate St., Eugene, OR 97403. Ph: 503/686-4414.