

THE NATIONAL LOGO EXCHANGE

Volume 3 Number 8

FORWARD 1001

April 1985

The Pause That Debugs

by
Harold Nelson

It happened during a recent seminar discussion. We were examining various methods of debugging Logo programs. Some users felt that STEP and TRACE are absolutely necessary tools for debugging and should be included in all Logo versions. But, as the discussion developed, it became clear that we had failed to promote the proper understanding and use of PAUSE as the dynamic, interactive debugging tool that it is.

STEP, TRACE, and PAUSE

STEP runs a line of a program and then waits for you to press a key before running the next line. It is sort of like watching a football game on a special video tape machine that stops after each play, allowing you to study it; when you press a button, the next play is run.

TRACE shows you an explanation of the result of each line of a program as it is being run. It is like watching the football game on the video machine accompanied by a good commentary, telling you what caused the results you saw.

Both of these are good debugging tools, but they do not allow you to interact with the program while it is running. When you can do that, powerful debugging is possible. Your students should experience this power.

PAUSE facilitates such debugging. Let's take a closer look.

CAUSING PAUSING

There are two ways to cause a pause while a Logo program is running: press the appropriate key(s), such as Control-Z for Logos on the Apple computer, or use PAUSE inside a procedure. Here is an example which uses both ways.

The COUNTDOWN procedure counts down from an input number, but it has a bug: it never stops.

```
TO COUNTDOWN :COUNT
PRINT :COUNT
COUNTDOWN :COUNT - 1
END
```

```
COUNTDOWN 5
```

```
5
```

```
4
```

```
3
```

```
2
```

```
1
```

```
0
```

```
-1
```

```
-2
```

```
...
```

Press Control-Z (or whatever) to make the procedure pause. Logo prints a message telling you where it is pausing, and waits for an instruction.

```
PAUSING... IN COUNTDOWN:
PRINT :COUNT
COUNTDOWN ?
```

Now it is possible to do a number of interesting things not possible with STEP or TRACE which can help with debugging. For example, you could verify the values of any active global or local variables, such as that of :COUNT.

```
COUNTDOWN ? PONS
COUNT IS -21 (or some other value)
COUNTDOWN ?
```

Now let's edit COUNTDOWN while it is pausing, and add PAUSE to it!

```
COUNTDOWN ? EDIT "COUNTDOWN
```

Change the procedure until it looks like the one on the following page, and leave the editor in the usual way. Note that you are still in the paused condition after the COUNTDOWN procedure is defined.

Pause continued

```
TO COUNTDOWN :COUNT
PAUSE      <--This line added
PRINT :COUNT
COUNTDOWN :COUNT - 1
END
```

```
COUNTDOWN DEFINED
COUNTDOWN ?
```

Next, let's reset the value of COUNT to 3, and then resume running the procedure (with CO), in order to catch the bug.

```
COUNTDOWN ? MAKE "COUNT 3
COUNTDOWN ? CO
PAUSING... IN COUNTDOWN:
PAUSE
COUNTDOWN ?
```

By entering CO after each COUNTDOWN ?, you can step through the program. Now, however, when the count goes through zero, you can identify the bug, edit the procedure once more, and replace the PAUSE with something like IF :COUNT < 0 [STOP].

This rather obvious bug serves as a simple example to show how PAUSE can be used. Think about what you have done. You have just edited and run a procedure inside of a pause of that very procedure in order to debug it! I think that this is really amazing.

TURTLE PAUSE

Now suppose you were improving a recursive interactive turtle graphics program, such as the following.

```
TO DRAWIT
SETHEADING TOWARD READCHAR
FORWARD 5
DRAWIT
END
```

```
TO TOWARD :DIRECTION
IF :DIRECTION = "U [OUTPUT 0]
IF :DIRECTION = "R [OUTPUT 90]
IF :DIRECTION = "D [OUTPUT 180]
IF :DIRECTION = "L [OUTPUT 270]
OUTPUT TOWARD READCHAR
END
```

Type it in and work with it a while. Then make it pause from the keyboard and move the turtle home.

```
PAUSING... IN DRAWIT:
SETHEADING TOWARD READCHAR
DRAWIT ?PENUP HOME PENDOWN
DRAWIT ?
```

This demonstrates that you can change the turtle's state without stopping and rerunning the program.

Now type CO. The program crashes and Logo prints a complaint. This is because this particular program was expecting a character when it continued. Here is how to take care of that. Type DRAWIT and then make it pause once more.

```
PAUSING... IN DRAWIT:
SETHEADING TOWARD READCHAR
DRAWIT ?
```

Now type CO and an input, such as "R. The program continues and the turtle immediately executes a move to the right.

```
DRAWIT ? CO "R
```

Now you may decide to add a key H to make the turtle return to the home position. Make the program pause from the keyboard and type EDIT "TOWARD. Add this line to the procedure and redefine it.

```
IF :DIRECTION = "H [PENUP HOME PENDOWN]
```

When you leave the editor, you may note that you have lost your original picture and that Logo has delivered a mild complaint, depending on your particular version of Logo. But the purpose of this example is to demonstrate the interactive debugging power, which is available to you the next time you are trying to fix up a more complex graphics program.

CONCLUDING THE PAUSE

In our seminar debugging discussion, we came to think of PAUSE as a sort of "time out" -- a special kind of interlude in which you can not only analyze the game so far, but during which you can also interact with and change the game. You can examine the values of variables, even local variables (inputs to procedures), assign and test new values, and inspect and edit the procedures being used.

Such powerful interactive debugging is well worth investing a little time to learn it and to teach it.

Harold Nelson is currently completing a book on advanced Logo programming concepts. He conducts Logo workshops around the world.

From the Editor

Logo and Sex

Seymour Papert observed that nearly half of the attendees at the Logo 84 conference last year were women. To an outsider, this might come as a surprise, since the computer field seems to be dominated by males. However, I feel that the large number of women was a manifestation of the Logo philosophy in action. "No threshold, no ceiling" should apply to everyone.

But is it really that way in the Logo classrooms?

The sexism issue is an important one. I recently took a workshop which sensitized me to how teachers react in different ways to male and female students.

One of the things suggested by the workshop instructor was that teachers tend to give detailed instructions to male students, but operate equipment or perform tasks for the female students, thus denying them the opportunity to learn by doing.

Since the workshop, I have been more aware of the impact of my words and actions in class. For example, I noticed that I would sometimes type at the Logo keyboards of the female students instead of giving them instructions. That was changed in a hurry!

I have asked my students to help me identify any other similar actions which put either males or females at a disadvantage.

Perhaps it would be worthwhile for all of us to "observe ourselves" as we interact with our male and female Logo students. Do we treat them differently? Do we allow access or deny opportunity? Do we encourage or discourage?

Here are a couple of recent articles well worth reading.

"The Computer Gender Gap," by Gordon Williams, Family Weekly magazine (Sunday newspaper insert), February 24, 1985, p. 7.

"Sexism in the Schoolroom of the '80's," by Myra and David Sadker, Psychology Today, March 1985, p. 54.

For a more comprehensive look at the equity issue as a whole, see The Computing Teacher of April 1984.

Let's make it "no threshold, no ceiling for everyone."

FD 100!



Tipps for Teachers

by Steve Tipps

Survey Sense II

When information is collected with a survey, questionnaire, or test, organizing and analyzing are important tasks. Collecting data, displaying it on a chart, finding the items with the most or fewest responses, and calculating the percentages of different categories have been discussed for the last two months. This month, data organizing tools are expanded to the high and low scores and the average. The idea of the "middle" of the information and its "spread" are also introduced.

MAKING ALLOWANCES

Interesting surveys need interesting topics. Money is always interesting to most people. How do you get it? How much do you get? What do you do with it? Doing a survey on allowances follows the same process as a face-to-face interview.

How much allowance do you get each week?
(Wait for an answer.)

Thank you for answering.

The program for this dialog is fairly simple, although a few extra directions help you get the right information. The tricky part is forming the list of allowances. This survey puts all the answers into the "ALLOWANCES list.

TO SURVEY

PRINT [HOW MUCH ALLOWANCE DO YOU GET EACH WEEK?]

PRINT [TYPE YOUR ANSWER IN DOLLARS AND CENTS]

PRINT [WITHOUT A DOLLAR SIGN.]

PRINT [THIS IS AN EXAMPLE:]

PRINT [2.50]

PRINT [NOW TYPE IN YOUR ALLOWANCE.]

MAKE "ALLOWANCES LPUT FIRST READLIST :ALLOWANCES

PRINT [THANK YOU FOR ANSWERING.]

END

At the beginning, the global variable "ALLOWANCES must be an empty list. This is only done once before any surveys are made.

MAKE "ALLOWANCES []

Tipps continued

Each SURVEY will add one item to the list. Of course, you could make separate surveys for different classes by defining separate procedures and variables names for each class.

```
MAKE "BLITMAN.ALLOWANCES [ ]
TO SURVEY.BLITMAN.CLASS
etc.
```

Survey forms can be as fancy as you want. On the allowance survey, a title page with money or a piggy bank is a nice addition. A more complete allowance survey might even ask for a distribution of the total spent into several categories: snacks, records, videos, movies, clothes, etc. Each of the categories could be stored in separate lists.

Now you want to organize, summarize, and display to show how much allowance is typical of the people in the survey. A couple of calculating tools can very handy.

LOOKING FOR THE EXTREMES

How can the list of allowance values be described? One way is to use the highest and lowest allowance. Here is a sample survey list. Look through the list and pick out the high and the low amounts. (Although amounts such as 3.00 were typed in when the SURVEY was taken, most Logo versions eliminate the zeroes when defining the ALLOWANCES list.)

```
ALLOWANCES IS [1. 2.5 1.5 2.25 4. 1.
1.25 3. 2.5 1.25 2.5 3. 2.75 2. 2.75 3. 2.25
2.5 3. 2.5 2.25 3. 2.75 2.5 3.]
```

Looking for the highest, you would probably say to yourself, "The first one is a dollar; that's the highest so far. The next is 2.50; that's higher. Now I'll use 2.50 to compare with the next one." The highest is selected by picking and comparing. Each time a higher value is found, it replaces the former high value.

Talking through this scanning process helps to understand how Logo could go about finding the high and low, because the process parallels the talk.

```
TO FIND.HIGH :LIST
IF :LIST = [ ] [ OP :HIGH ]
TEST :HIGH < FIRST :LIST
IFTRUE [ MAKE "HIGH FIRST :LIST ]
OUTPUT FIND.HIGH BUTFIRST :LIST
END
```

To begin, establish a low number as the first "HIGH value.

```
MAKE "HIGH 0
PRINT FIND.HIGH :ALLOWANCES
4.
```

The procedure to FIND.LOW is similar, only the initial value for "LOW must be set high to start.

COUNT FOR COUNT

Another way to organize the information is to count the number of times each value appears in the list. If you were doing this by hand, you would start at the low value and count how many there were. Then you would move to the next higher value and count again. You would keep doing this until you got to the highest value. That is exactly what Logo can do, too.

```
TO COUNT.EACH :LIST :TARGET
IF :TARGET > :HIGH [STOP]
MAKE WORD "C :TARGET 0
TYPE :TARGET TYPE [ \ \ \ ]
PRINT COUNT.EACH.LOOP :LIST
COUNT.EACH :LIST :TARGET + 0.25
END
```

```
TO COUNT.EACH.LOOP :LIST
IF :LIST = [ ] [OUTPUT THING WORD "C
:TARGET]
TEST :TARGET = FIRST :LIST
IFTRUE [MAKE WORD "C :TARGET (THING
WORD "C :TARGET) + 1]
OUTPUT COUNT.EACH.LOOP BUTFIRST :LIST
END
```

The new "trick" in these procedures is to create a variable name for counting each value. WORD "C :TARGET makes a variable name such as "C1.5 for counting the 1.50 values. The results look something like this. Use a cursor command such as SETCURSOR LIST 8 LAST CURSOR just before the PRINT line in the COUNT.EACH procedure if you want to have the counting results appear in a neat vertical line.

```
COUNT.EACH :ALLOWANCES :LOW
1. 2
1.25 2
1.5 1
...
...
4. 1
```

Tipps continued

A PICTURE IS WORTH A THOUSAND CHARTS

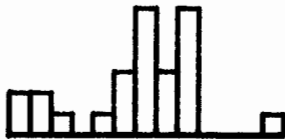
A graphic display would show even better the distribution of the allowance values. A BAR chart is easy to plot using values from COUNT.EACH.LOOP in the SHOW.EACH procedure.

```
TO SHOW.EACH :LIST :TARGET
  IF :TARGET > :HIGH [STOP]
  MAKE WORD "C :TARGET 0
  BAR 5 * COUNT.EACH.LOOP :LIST
  MOVE
  SHOW.EACH :LIST :TARGET + 0.25
  END
```

```
TO BAR :SIZE
  REPEAT 2 [ FD :SIZE RT 90 FD 5 RT 90 ]
  END
```

```
TO MOVE
  RT 90 FD 5 LT 90
  END
```

The bar chart looks like this.



The picture shows how far the allowances are spread out and indicates the most frequent allowances. It also gives a visual sense of the middle value. Some Logo versions make it possible to label the values on the graphics screen.

MIDDLING

When information is collected in categories equal in size, one way to summarize the data is with the middle scores. Actually, there are two different middle scores which can be used in certain circumstances. You may remember them from your Tests and Measurement courses.

The median is the score identified by counting halfway from the bottom to the top of all the scores. A median can be found quickly. For example, if the scores are [12 14 14 15 17 18 20 20], the median is 15 because it is the 5th of 9 scores.

The more common way to describe the middle is with the mean. The mean is the arithmetic average of scores calculated by adding all the scores together and dividing by the number of scores. The total of the nine scores above is 146, so the mean is 16.

Older children are usually familiar with averages from averaging grades. Although a calculator is useful for determining averages, Logo can be used for both collecting and calculating information. Its advantage is that information can be stored easily in the computer in lists and be subjected to several different analyses. Many calculators have no memory for data storage, or ways of keeping several different sets of information.

Writing a program to calculate the mean is fairly straightforward. Work through the list adding one item at a time to the total. Then divide the total by the number of items in the list.

```
TO AVERAGE :LIST
  MAKE "SUM 0
  OUTPUT ( TOTAL :LIST ) / COUNT :LIST
  END
```

```
TO TOTAL :LIST
  IF :LIST = [ ] [ OUTPUT :SUM ]
  MAKE "SUM :SUM + FIRST :LIST
  OUTPUT TOTAL BUTFIRST :LIST
  END
```

```
PRINT AVERAGE :ALLOWANCES
2.4
```

The way you handle the average procedure in your class will depend on the purposes of your lesson. For example, if you are working in sixth grade through high school mathematics, you can have students describe how to calculate the average and help them express the process in Logo. If you are using the procedure as a tool for a data collection, you can write the tool in a file for children to use. Whether to teach how to write the program or use it for analysis is not crucial; your goal for students is the important focus for your Logo work.

The bar chart provides a picture of how spread out the scores are. With a different survey you might get a different distribution. Looking at the spread is a very important idea with data. The standard deviation is a way of describing the spread of the scores. Next month, the calculation of the standard deviation will be introduced. For now, I'll leave it as a challenge for you and I'll share my method with you in May.

Tips continued

(By the way, the conversion of these procedures to Terrapin is straightforward. However, there is some kind of a rounding error present which fails to select all categories in COUNT.EACH. One way around this is to collect the allowance information in pennies to the nearest 25c. For example, 2.50 should be 250.)

Steve Tipps is the West Professor of Education at Midwestern State University in Wichita Falls, TX. He presents Logo workshops for school systems throughout the United States.

PAWS for the Turtle

Harold Nelson contributes this program for those whose Logo versions do not have a PAUSE command. Embed PAWS in your favorite procedure and study how it works. What limitations can you find?

```
TO PAWS
TYPE [PAUSING... ?]
PAWS.AUX READLIST
END
```

```
TO PAWS.AUX :INSTRUCTION.LIST
IF :INSTRUCTION.LIST = [CO] [STOP]
RUN :INSTRUCTION.LIST
TYPE [PAUSING... ?]
PAWS.AUX READLIST
END
```

For more suggestions on using PAWS or PAUSE for debugging, see Harold's article on page 1 of this issue.

Looking for Freebies?

Freebies is a magazine no teacher should be without! Published every month with a circulation of over 500,000, Freebies contains information on how to obtain free (or nearly free) items. Each issue has a special section for teachers. The cost? A mere \$6.97 per year. By the way, the NLX UVA Logo Class Penpal Network was featured in their March 1985 issue. Freebies, PO Box 20283, Santa Barbara, CA 93120. (805) 962-9135.

NLXionary

A Lectionary and Discussion of Logo Readings

by Griff Wigley

I remind readers that I am opening up this column to you for the purpose of ideological exchange on issues raised by current articles on Logo. I'll reserve my comments on the selections until I hear from you. Write to me at 918 College Street, Northfield, MN 55057.

"Logo and the Teaching of Problem Solving: A Call for a Moratorium," by Toby Tetenbaum and Thomas Mulkeen, Educational Technology, November 1984.

>The authors note that the excitement generated by Logo counts not so much from its perceived value as a programming language, but rather from its promises to enhance children's cognitive capacities and develop higher level cognitive skills which will generalize or transfer to other content areas. They challenge this view, comparing those who believe Logo teaches children how to think to those who claim that the study of Latin improves the mind through the discipline required to learn it. Research has failed to prove that there is any significant transfer from training on one task to success on another.

>They challenge the idea that the type of thinking employed when one solves a problem can be improved if one learns to "think" like a computer does when it solves a problem. They contend that the whole arena of problem solving is so huge and complicated that it is impossible to determine a single set of skills that can be identified as crucial to problem solving ability. They cite research that indicates problem solving can't be separated from subject matter content. They contend that Logo proponents have created unrealistic expectations regarding its ability to foster a generalized problem solving ability.

>Tetenbaum and Mulkeen refer heavily to the Bank Street study by Roy Pea and associates which indicated children who learned Logo didn't preplan prior to programming, didn't understand features of the language

NLXionary continued

(recursion, conditional statements, sequentiality) even though they used them, and failed to display better planning skills than the control group.

>Rather than throwing out Logo, the authors suggest a moratorium on promoting it as a generalized problem solving model. They indicate that it might yet prove to be so, if educators create two conditions that enhance the transfer of knowledge and skills:

1. Some of the processes or knowledge used in one task should be identical to those learned while engaging in another.

2. Learners should be made aware specifically of the skills involved in the task, as well as the specific content of the task.

>Pea has evidently suggested that instruction in thinking skills in conjunction with Logo might be beneficial, despite his description of Logo as "Wheaties for the mind." The authors, however, clearly doubt if teaching problem solving through the teaching of programming is the most efficient way to bring it about.

>"Before tens of thousands more children are taught Logo, it seems advisable to give serious consideration to its purpose."

>Do we all agree?

(Ed. Note: If you have access to CompuServe, you may find the lively discussion about this article on TeleCUE to be of interest. Type GO WCC-21 after logging on, and read Logo messages from #1953 to #2052.)

Griff Wigley is a co-owner of Family Computing, Inc., a facilitator in the Faribault (MN) Public Schools, and a school board member of Prairie Creek Community School.

Turtle Tips

by
Donna Lanyi
and Jane Toth

By this time of the year, your students are creating some interesting procedures. Many of their ideas are being saved on disks. Sharing is a great way to motivate students. Let's take a look at ways to find time for students to share their ideas with their classmates.

TEACHER HINTS

Planning a 20 - 30 minute sharing time per week is an excellent habit. This sharing time can be used for multiple purposes.

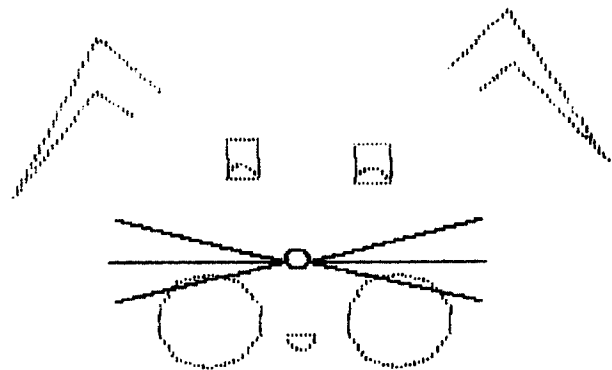
First, it enables the class members to see what others are doing. When a student completes a Logo project, your classroom doesn't have to be interrupted for "show and tell." Everyone knows they will get to see the project during sharing time.

Second, the sharing period can be used as a catalyst for other students who have had "idea stagnation." We have seen a number of students get exciting ideas for spinoff projects from something a classmate reports.

A third purpose of a sharing time is to allow students an opportunity for constructive criticism of each other's work. The teacher should direct the discussion to keep it positive. As a result of these remarks, additional improvements, additions, and extensions often suggest themselves.

PROCEDURE OF THE MONTH

Hop to it! Have your class design an Easter parade of spring procedures. Use our bunny procedure as a motivating idea.



If you would like a free listing of this procedure, send us a self addressed stamped envelope.

We invite elementary teachers and students to send procedures or ideas to be shared. Donna Lanyi and Jane Toth, Beall Avenue School, 716 Beall Avenue, Wooster, OH 44691.

Donna Lanyi and Jane Toth use Logo in their elementary classrooms in the Wooster (OH) City Schools.

Listful Thinking

by
Glen Bull and Paula Cochran

Word Machines

Spring has finally arrived at Mr. Jefferson's University. Mr. Jefferson was something of a tinkerer, continually designing gadgets and architectural improvements for Monticello. We have been tinkering with Logo machines.

LOGO MACHINES FOR LANGUAGE TINKERING

Judah Schwartz, co-director of the Educational Technology Center at Harvard, compares Logo to a "machine shop" in which any tool can be built, but only with considerable time and skill (see NLXionary, March NLX, 1985). For the past year, we have emphasized use of Logo tools to teach content-specific concepts in language arts.

In this month's column, straight from our Logo machine shop, we present the Great American Word Machines. Schwartz says (and we agree) that using Logo can be like trying to make a pair of pliers from a blank piece of metal. In this column we present some intermediate tools. These tools let you stock your own machine shop with pliers, screwdrivers, and other tools. We hope this lets you begin teaching language arts, instead of having to spend time making Logo tools to teach language arts.

Children are inquisitive about language. A four-year-old may experiment with variations of a word in a sing-song voice. Children sometimes invent their own "secret" languages. Piglatin is an example which children re-invent in each generation. This early fascination with language is rarely incorporated into formal grammar instruction in the classroom. We need ways to use children's natural playfulness and enthusiasm for language in English class.

By way of contrast, here is a typical lesson from a sixth grade English text:

In the sentences below, change each adjective to an adverb. Tell how each adverb is formed.

1. Tom was _____ answering his NLX mail.
(eager)
2. Steve was _____ writing his column.
(quick)

The object of the lesson is to learn how adverbs are constructed. By adding -ly endings, the student is supposed to deduce a rule about English adverbs. How can we construct a Logo machine which will supplement and extend this lesson?

WHAT'S IN THE MACHINE SHOP?

Our Logo machine shop comes equipped with four types of tools.

- 1) separators
- 2) joiners
- 3) omitters
- 4) testers

These tools are used to build Logo machines, for the exploration of different subjects. We're going to look at some word machines.

The Logo command WORD joins any two parts:

```
PRINT WORD "EAGER "LY
EAGERLY
```

Here's a Logo machine which uses the command WORD to convert adjectives into adverbs.

```
TO ADVERB :ADJECTIVE
OUTPUT WORD :ADJECTIVE "LY
END
```

```
PRINT ADVERB "EAGER
EAGERLY
PRINT ADVERB "QUICK
QUICKLY
```

CAUTION! WORDS UNDER CONSTRUCTION

There are several ways to use this Logo machine to supplement the textbook assignment. The teacher could ask students to guess what the machine is doing. In this case, the children are being asked to deduce the rule which the machine uses to create adverbs. This could be a group activity in which the teacher types in sample words. Older students could be given a page of sample words to test on their own. A page of sample words from the assignment might look like this:

IN	OUT
eager	eagerly
soft	_____
quiet	_____
sure	_____
sudden	_____

Listful Thinking continued

Another way to use this machine would be to give children the rule, then let them explore the boundaries of the rule using the Logo machine. The assignment might look like this:

Here is a rule for making adverbs from adjectives:

Add -ly to some adjectives to form an adverb.

slow --> slow + ly = slowly

Use the Logo ADVERB machine to test this rule. Which of these words make correctly formed adverbs, and which do not?

IN	OUT
nice	nicely
abrupt	_____
frantic	_____
sincere	_____
think	_____

TINKERING AND TESTING

It's possible to turn the idea around, and construct a Logo machine that checks to see if the last two letters are -ly. This machine uses a Logo tool that separates the last two letters from the word. (See the end of the column for LAST.2.LETTERS.)

```
PRINT LAST.TWO.LETTERS "NICELY
LY
PRINT LAST.TWO.LETTERS "FROG
OG
```

The LAST.2.LETTERS procedure is used here to construct a machine that tests for adverbs ending in -ly.

```
TO ADVERB? :ITEM
OUTPUT (LAST.2.LETTERS :ITEM) = "LY
END
```

It works like this:

```
PRINT ADVERB? "NICELY
TRUE
PRINT ADVERB? "FROG
FALSE
```

Of course, the machine won't work perfectly.

```
PRINT ADVERB? "SALLY
TRUE
```

A child will often discover these cases, and ask why the machine doesn't work. This provides a good opportunity for a discussion about the kinds of knowledge people need to make distinctions between nouns and adverbs. What else would the ADVERB? machine need to work better?

THE ALL-PURPOSE CHECKER CHECKS ADJECTIVES

The same approach used for the "ly" checker can be used to build a general purpose tester. This checker could be used to test adjective suffixes, for example.

First, make a list of adjective suffixes. We use "ful" and "ive" but you could make up your own list.

This procedure outputs the list of adjective suffixes.

```
TO ADJ.SUFFIX
OUTPUT [FUL IVE]
END
```

The procedure to test for adjective suffixes looks like this. The LAST.3.LETTERS procedure is at the end of the column. (If you are using Terrapin Logo, use the CHECK procedure at the end of the column.)

```
TO CHECK :ITEM
TEST MEMBERP LAST.3.LETTERS :ITEM ADJ.SUFFIX
IFTRUE [OUTPUT "TRUE]
OUTPUT "FALSE
END
```

This particular checker is used like this:

```
PRINT CHECK "THANKFUL
TRUE
PRINT CHECK "NICELY
FALSE
PRINT CHECK "MASSIVE
TRUE
```

This suffix checker looks for adjective suffixes, so it picks out "thankful" and "massive", but not the adverb "nicely." The suffix tester could be used in an assignment like this:

Here is a list of words. Test each word with the Logo machine CHECK. Then write down the result in the blank beside the word.

Listful Thinking continued

IN	OUT
thankful	True
kindness	_____
massive	_____
nicely	_____
darkness	_____
permissive	_____
bountiful	_____

After checking each word, decide what kind of word the suffix tester is identifying: NOUN, ADJECTIVE or ADVERB.

These ideas are designed to enhance traditional pencil-and-paper language activities. Logo word machines can help students explore language and examine rules by taking advantage of their natural curiosity. Next month we will look at some other Logo language machines that use separators, joiners, omitters, and testers.

PROCEDURES

```
TO PIGLATIN :ITEM
OP ( WORD BUTFIRST :ITEM FIRST :ITEM "A )
END
```

```
TO LAST.2.LETTERS :ITEM
OP WORD NEXT.TO.LAST :ITEM LAST :ITEM
END
```

```
TO NEXT.TO.LAST :LETTER
OP LAST BUTLAST :LETTER
END
```

```
TO LAST.3.LETTERS :ITEM
OP WORD THIRD.FROM.LAST :ITEM
LAST.TWO.LETTERS :ITEM
END
```

```
TO THIRD.FROM.LAST :LETTER
OP LAST BUTLAST BUTLAST :LETTER
END
```

If you are using Terrapin Logo for the Apple, or Commodore Logo, use this CHECK procedure.

```
TO CHECK :ITEM
TEST MEMBER? LAST.3.LETTERS :ITEM ADJ.SUFFIX
IFTRUE OUTPUT "TRUE
OUTPUT "FALSE
END
```

If you are using Version 1 of Terrapin Logo for the Apple, you will also need the MEMBER? procedure below.

```
TO MEMBER? :ITEM :LIST
IF :LIST = [ ] OUTPUT "FALSE
IF :LIST = " OUTPUT "FALSE
TEST :ITEM = FIRST :LIST
IFTRUE OUTPUT "TRUE
OUTPUT MEMBER? :ITEM BUTFIRST :LIST
END
```

Glen Bull is a professor in the University of Virginia's School of Education, and teaches Logo courses at both the graduate and undergraduate level. Paula Cochran is a reformed English major who studied linguistics at the University of Cambridge. She is now a speech-language pathologist working with language-disabled children and Logo.

Logo Notes

The following Logo versions are available from The Programmer's Shop. Dr. Logo at \$119, Experlogo for \$140, Commodore Logo for \$65, TI Logo at \$99, PC Logo for \$135, Terrapin Logo for \$89, and TLC Logo for \$95. The Programmer's Shop, 128 Rockland Street, Hanover, MA 02339. (800) 421-8006.

Available from Triton Products: Early Logo Learning Fun (#AAFH), a TI 99/4A cartridge for preschool children by Spinnaker at \$11.95, and the TI Logo II (#AEBE) cartridge for the TI 99/4A at \$99.95 plus \$3.00 for shipping. A catalog listing many other TI 99/4A products is also available. Triton Products, PO Box 8123, San Francisco, CA 94128.

Miles Kimball Company, a mail order firm, sells a "cast iron" turtle string holder complete with a large ball of red twine. We threaded the twine so it came out of the bottom of the turtle, and used it as a concrete drawing object. Thumbtacks fix the string in place as the turtle maneuvers over a cardboard surface. Item # 7906, String Holder Turtle, \$3.49 plus \$1.95 (ouch!) shipping. Miles Kimball Company, 41 West 8th Avenue, Oshkosh, WI 54906. (414) 231-3800.

Logo Disserts:

Dissertations Dealing with Logo

by Barbara Elias

What aspects of Logo instruction may serve to prepare children for our more technically oriented society? Problem solving, motivation to learn, and attitudes were investigated in a study of the implementation of a Logo programming course for fourth grade children. Eight of fifteen local fourth grade classes were used as subjects, while the remaining seven served as the controls.

Classroom teachers who provided the instruction in Logo were trained in a summer inservice Logo programming course at the University of Virginia. The course placed special emphasis on the Logo philosophy, and stressed the importance of encouraging exploration with Logo. It incorporated such concepts as modularity and debugging.

Once school began and classes started, teachers were expected to model course behaviors for the children involved. Teacher inservice continued throughout the school year and the inservice instructors worked closely with the teachers in developing classroom activities and curriculum ideas.

Teacher implementation of the Logo course was expected to vary because teachers were free to choose the most appropriate means for their individual classrooms. Generally, children worked in pairs during the 45 hours of class time set aside for the computer in the year long study.

The following three tests were administered at the beginning and at the end of the school year: the Cognitive Abilities Test (Nonverbal form), the Fennema-Sherman Mathematics Attitudes Scales, and the Scale of Intrinsic versus Extrinsic Orientation in the Classroom.

Results indicate that the Logo programming experience had a positive influence on problem solving abilities and on attitudes toward learning mathematics. Even though the differences were small, they represent potentially important differences between children who have had Logo programming and those who have not. While these results look promising, it is recommended that this study be replicated and extended, that Logo be integrated into the mathematics curriculum,

and that the importance of continuing teacher inservice support to the success of a Logo project be recognized.

>>>Evans, Jr., H. R. (1984). Some Effects of Logo Programming Instruction with Fourth Grade Children. (Doctoral dissertation, University of Virginia, 1984).

(Ed. Note: Each month, Barbara Elias highlights a dissertation or thesis dealing with Logo. She is accepting copies of recent research results for the NLX Dissertation and Thesis Repository. Reports may be mailed to her in care of NLX, PO Box 5341, Charlottesville, VA 22905.)

Barbara Elias is an assistant professor in the Education Department of Virginia State University in Petersburg, VA, and a doctoral candidate at the University of Virginia.

Logo Announcements

Terrapin is introducing an upgrade for their Apple computer version of Logo. Terrapin Logo 3.0 is really two languages in one. When booted on an Apple computer, the program identifies the model and amount of resident memory, and then loads either a 64K version, or one which can address a full 128K, if available. Thus, with a single Logo disk, you can take advantage of the full memory capabilities of every computer in the Apple II family. In addition to addressing more memory, Terrapin 3.0 also loads much faster than previous versions. The suggested list price is \$99.95. Owners of Terrapin Logo 1.3 or 2.0 may upgrade for a small fee. For more information, write to Bob Paolucci, Terrapin, 222 Third Street, Cambridge, MA 02142, or call (617) 492-8816.

Terrapin also announces the release of a Logo cartridge for the Commodore Plus/4, making available roughly twice the memory space of Commodore Logo at a suggested list price of \$74.95.

ExperTelligence Inc. announces Exper-Logo, Logo for the Macintosh. Included are many advanced features not found in other Logo versions. Additionally, ExperLogo is compiled, resulting in fast programs. The suggested list price is \$149.95. ExperTelligence, 559 San Ysidro Road, Santa Barbara, CA 93108. (805) 969-7874.

Teacher to Teacher

The Turtle Goes to Kindergarten

A guest review
by Jan Cardwell

Turtles are popping up everywhere! Since Logo went public, turtles have gained rapidly in popularity and in possibility. The momentum of the race continues forward as turtles are propelled into the kindergarten and introduced to the very young.

What better environment could there be for the Logo turtle than the kindergarten? Renown for their curiosity and willingness to explore, kindergartners mirror Seymour Papert's vision of Logo in Mindstorms.

For educators who have not had the "free" time to familiarize themselves with Papert's work, Donna Bearden and Kathleen Martin have the answers in a clear, concise, and easy to read book, The Turtle Goes to Kindergarten. Instead of bringing the student up to the level of the computer, Donna and Kathleen have prepared ways to bring Logo to the level of the kindergartner.

LEVELS OF LOGO

In order to broaden the Logo capabilities of such beginning readers, the authors have developed programs for single stroke commands. The three levels presented increase in degree of difficulty, and correlate to a kindergartner's interest and developmental abilities.

Realizing that young children learn best by doing, the authors have included activities which help the child experience abstract concepts in a very concrete way.

Especially inspiring is the section entitled "Pre-Logo Activities." Here, suggestions are made which immerse the kindergartner in learning. Children become turtles, making discoveries as turtles. These Logo lessons are not forgotten when the students are guided by self discovery.

WHERE'S THE BEEF?

My favorite activity was one of the "Playing Turtle" games. Students take turns pretending to be the turtle, as classmates give verbal commands directing the turtle to -- the turtle food!

All kindergarten teachers know what a motivator this can be! What would kindergarten turtles eat? When I asked my students, they responded with answers ranging from flies (yuk!) to jelly beans!

Donna and Kathleen certainly know the way to a kindergartner's heart (and stomach) as well as the way to the brain!

Throughout the book, teachers are encouraged to integrate Logo learning into the kindergarten classroom environment. In fact, the entire second part of the book is devoted to programs and techniques which teach basic kindergarten curriculum concepts (spatial relations, colors, arithmetic) through the use of Logo and the computer. The programming work has already been done.

IN SUMMARY

In conclusion, I must mention that the most compelling aspect of The Turtle Goes to Kindergarten is the way the authors encourage the reader to e x p a n d upon the ideas presented. Not only do Donna and Kathleen teach us how to incorporate Logo into the kindergarten, but they also inspire us to do some learning, too!

The Turtle Goes to Kindergarten is available from the authors for \$5.00 plus \$1.00 shipping and handling. Martin - Bearden, Inc., PO Box 337, Grapevine, TX 76051.

Jan Cardwell, a former preschool teacher, is a program coordinator for the Science Museum of Virginia. She has published several Logo teaching tips in national educational periodicals.

Free Logo Materials

The Cotting School for Handicapped Children in Boston has a limited number of free booklets on Logo activities available. Logo for Young Children and Children with Cognitive or Physical Handicaps: Simplified Versions of Logo by Susan Jo Russell offers four modified Logo versions and suggestions for their use. Logo Games and Activities by Susan Jo Russell and Eileen Kiernan includes games and other activities for 6 to 10 year old children, and for children with special cognitive or physical needs. Send a request on your letterhead to Development Office, Cotting School for Handicapped Children, 241 St. Botolph Street, Boston, MA 02115. This offer is good only while the supply lasts.

前 100

by
Hillel
Weintraub

Last month, I introduced the Bynas Division of UNY Company and their work in the spread of Logo throughout Japan, and told of their Logo Journal, a monthly newsletter devoted to keeping a growing Japanese Logo community informed about the educational and commercial activities of Logo around the world. This month, I would like to introduce a second project presently under development by Hiroyoshi Goto of Bynas: the formation of the Japanese Logo Research Group, an organization composed mainly of educators which has been involved in Logo developments in Japan.

The first work of the group was to develop a set of standard Logo specifications for use with Japanese primary school students. The tentative list of specifications will be announced this May, and the final set in December. The purpose of creating such a list is to "make Logo easily accessible so that programs can be accumulated and shared." In order to give Logo the ability to operate on several types of machines, this group is planning to make their language specifications public and encourage all Logo producers to adopt it.

In my experience using Logo (which is almost totally with 10 - 12th graders working with words and lists), I have developed some strong feelings about useful primitives, hardware capabilities, and the like. Here are a couple of the ideas I plan to share with the research group.

1. A TRACE command is extremely important. A primitive is easier to use than a TRACE procedure which needs to be defined by the user. One suggested improvement would be to have the program text output printed on the screen in inverse letters. This would make it easier to separate what is part of TRACE and what is output of the procedure in operation. Even better would be to have the TRACE output shown on a separate "page" especially for debugging. Thus, the TRACE instructions could be overlaid without disturbing the actual program operation. This would emphasize the importance of planning and debugging in Logo. (Ed. Note: For a different perspective on TRACE, see Harold Nelson's article elsewhere in this issue.)

2. Many students do not use the error message fully in the debugging process. I find that my students do not spend much time studying Logo complaints before they type ED. Once they type ED, the message is gone forever. In Japan especially, the compulsion to get rid of errors before they bite you is ingrained in students from the time they can pick up a pencil. It is a fact that some students here write with a pencil in one hand and an eraser in the other! I would suggest an editing system which does either of the following things:

a. Has no separate editor and thus doesn't erase error messages when displaying a procedure for editing.

b. Has an editing mode which displays the error message in a special box somewhere.

c. Has a "bug buffer" which recalls the last error message any time upon pressing a key, even when the user is in the editor.

d. Uses a special debugging "page" (as mentioned in 1 above) which could be overlaid with the error message.

Preservation of the error message would reinforce one of the main teaching points of Logo: to use our mistakes to help us learn!

These are a few of the ideas I plan to present to the Japanese Logo Research Group. How about you? Do you have anything to suggest which I could pass along for them to consider? If so, send your ideas to me at Doshisha International High School, Tatara, Tanabe-cho, Tsuzuki-gun, Kyoto-fu 610-03 Japan.

Hillel Weintraub is a teacher at Doshisha International High School in Kyoyo-fu, Japan, and is the president of the Society for Microcomputing in Life and Education (SMILE).

The New Microquests

Microquests, the popular monthly Logo publication from Kathleen Martin and Donna Bearden, will have a new look in the fall. Kathleen reports that they are changing the format to a quarterly 50-page booklet which will use a project approach to promote thinking skills, facilitate curriculum integration, and provide student encouragement and challenges. Disks will be available for each issue. Donna and Kathleen produce work of the highest quality, and this series of booklets will be no exception. For more information, write Martin-Bearden, Inc., PO Box 337, Grapevine, TX 76051.

Q and A

by
Jim McCauley

Q. I have managed to define an "inaccessible procedure" in Apple Logo:

```
TO "1
FD 100 RT 45 BK 20 FD 20 LT 45 BK 100
RT 90 FD 20 BK 40 FD 20 LT 90
END
```

I can't get it to run! And the only way I can change it in the editor and redefine it with Control-C is to put the quotation mark (") in front of the name (1) each time. What's going on here?

<[-?->

A. Usually, when you try to define something in any of the Logo versions from Logo Computer Systems Incorporated (LCSI), one of two things happen. If the name that you choose for the procedure happens to be the name of something that is already defined (either a primitive or a procedure), you get an error message as soon as you type in the "defining line."

```
TO FORWARD :STEPS
```

```
FORWARD IS A PRIMITIVE
```

If you choose the name of an already-defined procedure (say its name is FOO), you get:

```
FOO IS ALREADY DEFINED
```

If you choose as a procedure name something that is not yet defined, you are permitted to proceed normally.

OUT OF THE ORDINARY

You have uncovered an unusual situation. Numbers and certain other characters (control characters, for example) have special meanings in Logo, and they are, in a sense, reserved for use by the system.

However, they are not protected against use as procedure names in the way that primitives and defined procedures are.

The Apple Logo Reference Manual mentions "formal Logo" on page V. It is a section that bears scrutiny. In the more normal mode of use, Apple Logo is "relaxed." That is, one can take various shortcuts.

In formal Logo, however, any object that is not quoted or does not begin with a colon (as variable names and values do) is assumed to be the name of a primitive or procedure. Therefore, if we were obliged to program in formal Logo, the favorite saying of this newsletter's editor would appear:

```
FORWARD "100
```

Apple Logo recognizes this command, and carries it out.

If you really want to run your 1 procedure, here is one way.

```
TO RUNIT :WHAT
RUNNER BUTFIRST TEXT :WHAT
END
```

```
TO RUNNER :RUN.LIST
IF EMPTY? :RUN.LIST [STOP]
RUN FIRST :RUN.LIST
RUNNER BUTFIRST :RUN.LIST
END
```

```
RUNIT 1
```

Whew!

(Ed. Note: If you have a question about Logo programming techniques or the thinking which leads to Logo programming, send it to NLX Q & A, PO Box 5341, Charlottesville, VA 22905. Each month, Jim McCauley answers selected questions in this column.)

Jim McCauley is a Coordinator of Computer Education for the Santa Clara (CA) County Office of Education, and has written Logo articles for many national publications.

Welcome to the French Logo Users Group

The French Logo Users Group is now established! They publish a most informative 36-page periodical called LOGon, which contains articles, reviews, listings, and other interesting information, all in French. For more information, write Association Française des Utilisateurs de Logo, 12 rue de la Montagne Ste Genevieve, Paris, France 75005.

What If

by
Michael Tempel
and Marie Bourgoin

In March's NLXual Challenges column, Robs Muir presented our challenge of writing IF in Logo. Here are the first three versions we came up with at LCSl.

```
TO MICHAEL'S.IF :CONDITION :TRUE.LIST
:FALSE.LIST
GO :CONDITION
LABEL "TRUE RUN :TRUE.LIST STOP
LABEL "FALSE RUN :FALSE.LIST
END
```

Since this abomination uses GO and LABEL, we were determined to find something better.

```
TO YVES'.IF :COND :TLIST :FLIST
MAKE "TRUE :TLIST
MAKE "FALSE :FLIST
RUN THING :COND
END
```

This was quickly converted into an elegant one liner.

```
TO BRIAN'S.IF :COND :TRUE :FALSE
RUN THING :COND
END
```

These IFs work the same way as the Logo primitive, with two differences. They each require three inputs. Also, infix operations must be enclosed with parentheses.

```
BRIAN'S.IF (2=2) [PRINT "YES][PRINT "NO]
YES
BRIAN'S.IF MEMBERP "A [A E I O U][PRINT
[WE HAVE HERE A VOWEL.]] []
WE HAVE HERE A VOWEL.
BRIAN'S.IF MEMBERP "X [A E I O U][PRINT
[WE HAVE HERE A VOWEL.]] []
```

In the last example above, nothing happens because we have used an empty list for the "false" consequence.

TRUE WORD, FALSE WORD

An interesting feature of Logo is revealed by these procedures. The Logo predicates (MEMBERP, EQUALP, =, and the like) output either TRUE or FALSE; but TRUE and FALSE are ordinary Logo words and can be manipulated as such.

```
PRINT WORD (EQUALP "APPLE "IBM) (6>1)
FALSETRUE
```

```
PRINT SENTENCE "NIAGARA (WORD (BUTLAST
BUTLAST 1=2) "LS)
NIAGARA FALLS
```

```
MAKE "FALSE [THIS IS NOT TRUE.]
PRINT THING (4=3)
THIS IS NOT TRUE.
```

Next month, we will reveal the rest of our IFs and some of those sent in by NLX readers. Meanwhile, if you have an IF, send it to Michael Tempel, Logo Computer Systems Incorporated, 555 West 57th Street, Suite 1236, New York, NY 10019.

Both authors are employed by LCSl. Michael Tempel is the Director of Training and Mario Bourgoin is a Programmer / Analyst.

NLXual Challenges

by
Robs Muir

With only a few sterling exceptions, the current literature about the Logo family of languages is preoccupied with children's experiences. Nowhere is this more obvious than in the catalogs of "educational" suppliers, where scores of books highlight the language "designed for children." Indeed, the original intent of the Logo project at MIT was to create a computer language / environment that had "no threshold and no ceiling" suitable for younger children and "naive users" as well as sophisticated computer scientists. Current publications about Logo often choose to emphasize the simplicity of Logo to the exclusion of the more advanced aspects of the language.

Logo has powerful implications for "mature" computer users. Evidence for this can be found in such texts as Turtle Geometry by Abelson and diSessa, Thinking About [TLC] Logo by Allen et al., the MIT Logo project memos, and the body of literature about LISP - the petri dish for the germ of Logo. Despite these outstanding works, there is little evidence that Logo is being taken seriously in higher education, Tom Lough's editorial in the March NLX notwithstanding. There are few exceptions.

Dr. James Armentage, of the Claremont Graduate School, is currently using both Logo and Pascal as notational languages in his graduate level course in Information and Computer Science. The school's degree program is tailored to professionals already in

NLXual Challenges continued

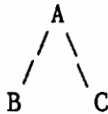
the field of information management who are interested in doing postgraduate work. Jim's course, entitled "Data Structures," includes Pascal only as a concession to the strong industry focus of some of the students. Jim has found Logo to provide a list processing notation ideally suited for exploration of complex concepts in data structure.

LOGO HOMEWORK CHALLENGE

This month's challenge is similar to one of the first assignments given to Jim's students. They are asked to go to the micro-computer lab, read the reference manuals on Logo, and begin programming. This is most likely their first exposure to Logo. Here is their assignment (and yours): Write a turtle graphics program that will represent a list of objects in a visual tree structure.

Allow me to illustrate...

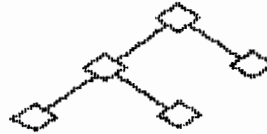
For the purposes of this challenge, let's select a list notation which has three elements: [node left.branch right.branch]. For example, the simple list [A B C] could be represented as



A more complex list like [A [B C D] E] could be represented as



With this quick background about lists, write a program that displays a list of non-primitive objects in this three-part format as a visual representation. You might choose to use little boxes to represent the list objects; this will be easier for the turtle to handle. It might also be possible to use the textscreen for this task, although this is beyond the scope of Armentage's original assignment.



Here are a couple of lists to use for testing your program.

```
[TALLER? BETTY [TALLER? TOM ROBS]]
[A [B C D] [E F [G H I]]]
```

In passing, you might notice that this process is similar to the format of prefix arithmetic operations. For example, compare the operation OUTPUT (PRODUCT 2 (SUM 3 4)) with the TALLER? list above.

Send your working solution (or even a valiant attempt) along with a self addressed stamped envelope to:

NLXual Challenges
 Attn: Robs Muir
 1688 Denver Avenue
 Claremont, CA 91711

After compiling the contributed solutions, we will select several of the most interesting or unusual ones and redistribute them to each contributor.

 Robs Muir is a teacher in the Claremont (CA) Unified School District and Claremont Graduate School.

The National Logo Exchange, copyright (c) 1985 by Posy Publications, a part of The Posy Collection, all rights reserved. Published monthly, September through May, \$25 (US) per year, mailed first class from Charlottesville, VA. \$5 (US) additional per year for addresses outside the USA, Canada, and Mexico. The opinions expressed by the authors are not necessarily those of The National Logo Exchange. Permission is granted for libraries and others registered with the Copyright Clearance Center (CCC) to photocopy articles herein for the flat fee of \$2 per copy of each article. Payment should be sent directly to CCC, 21 Congress Street, Salem, MA 01971. Address other correspondence to: The National Logo Exchange, Box 5341, Charlottesville, VA 22905. ISSN 0734-1717

Editor.....Tom Lough

NLXTRA

A number of school systems are taking advantage of our NLX discounts for multiple subscriptions mailed to a single address. Here is a listing of the discounts available for addresses in the United States. Persons ordering from other countries should write for additional information.

Number of subscriptions to a single address	Discount
10 - 19	10%
20 - 39	20%
over 40	30%

The following institutions offer introductory Logo courses for the summer session.

Fairfield University, Attn: Dr. John Schurdak, North Benson Road, Fairfield, CT 06430. (203) 255-5411.

Fresno Pacific College, Attn: Dr. Arthur J. Wiebe, 1717 South Chestnut Avenue, Fresno, CA 93702.

Millersville University, Attn: Dr. Edward Beardslee, Math and Computer Science Department, Millersville, PA 17551. (717) 872-3781.

National College of Education, Attn: Dr. Sandra Turner, 2840 Sheridan Road, Evanston, IL 60201. (312) 256-5150 x 418.

Northern Arizona University, Attn: John W. Glenn, Box 4117, Flagstaff, AZ 86011. (602) 523-4212.

University of Alaska, Attn: Charles K. Ray, 311 Eielson Building, Fairbanks, AK 99701. (907) 474-7021.

University of Houston, Attn: Dr. Sue Brown, 2700 Bay Area Boulevard, Houston, TX 77058. (713) 488-9494 and University of Houston, Attn: Dr. Barbara Sadowski, College of Education, Houston, TX 77004. (713) 749-7103.

Webster University, Attn: Dr. Andrea Rothbart, 470 East Lockwood, Webster Grove, MO 63119. (314) 968-7490.

Instructional Strategies in Logo, a 3 quarter hour course, is offered by Oregon State University during the summer session. For more information, write Dr. Margaret Moore, Weniger 253, Oregon State University, Corvallis, OR 97331. (503) 754-4031.

Come to the University of Virginia this summer for Logo courses. Glen Bull and Paula Cochran teach a 1 credit hour introductory Logo course June 12 - 18, and a 3 credit hour course on using Logo in the content areas, June 19 - July 11. For more information, write to Glen Bull, University of Virginia, 109 New Cabell Hall, Charlottesville, VA 22901. (804) 924-7107.

Dan and Molly Watt and Tony Stavelly present their 1985 Logo Institute at Keene State College in New Hampshire July 1 through 19. This popular program of study is designed for educators at all levels of Logo experience. Both theory and applications are studied through a variety of activities. Graduate credit is available. This is one of the more exciting summer Logo opportunities available. For a free brochure, write to Howard Croteau, Summer School Director, Keene State College, Keene, NH 03431. (603) 352-1909.

Logo Computer Systems Incorporated (LCSI) announces two series of Logo workshops in the New York City area. The first concentrates on Logo and the curriculum, and is presented on Thursday afternoons at Packer Collegiate Institute in Brooklyn, starting April 11 and running through May 30. The second deals with Sprite Logo, and meets Fridays at the Convent of the Sacred Heart in Manhattan's East Side from April 12 through May 31. For more information, contact Michael Tempel at LCSI, 555 West 57th Street, Suite 1236, New York, NY 10019. (212) 765-4780.

Two graduate courses on Logo and early childhood will be taught at Oakland University (MI) this summer. An introductory course and one for experienced Logo users both begin in late July. For more information, contact Logo Information For Teachers (LIFT), PO Box 5396, Plymouth, MI 48170.

A summer course entitled Logo Tools for Learning Mathematics is offered by Carleton College. For more information, write Office of Summer Academic Programs, Carleton College, Northfield, MN 55057. (507) 663-4301.

Logo Information For Teachers (LIFT) offers active membership to anyone interested in support for Logo teachers and teacher trainers. Members receive a membership directory, a listing of Logo research projects by members, and quarterly updates. To join, send \$10.00 to LIFT, PO Box 5396, Plymouth, MI 48170.

The Educational Computer Consortium of Ohio (ECCO) announces its spring Logo conference to be held Friday, May 3, at Mayfield School in Cleveland. Featured on the program is Joyce Tobias, presently associated with Seymour Papert's Microworlds project. An extensive schedule is planned, with activities for Logo users from beginner to expert. For more information, write to ECCO Logo Conference, 1123 S. O. M. Center Road, Cleveland, OH 44124, or call (216) 291-5525.

The 3rd annual Logo Information For Teachers (LIFT) conference is scheduled for Friday, May 3, at the University of Michigan, from 8:30 a.m. to 4:00 p.m. Co-sponsored by the U of M's Professional Development Office, the conference features Tony Stavely and other nationally known Logo workers. For more information, write LIFT Conference, PO Box 5396, Plymouth, MI 48170.

"Logo and Problem Solving," a 3 credit graduate course, is scheduled to be taught at Wright State University in Dayton, OH, beginning June 10. The course provides an introduction to Logo with a major focus on the problem solving processes of the language. Dr. Carl Benner is the instructor. For more information, write Bonnie K. Mathies, Program Coordinator, 244 Millett Hall, Wright State University, Dayton, OH 45435, or call (513) 873-2509.

Logo and the Learning Environment, a 4 quarter hour course, is offered this summer at Eastern Washington University. For more details, write Stanley Robinson, Department of Computer Science, Cheney, WA 99004. (509) 458-6260.

The Graduate School of Education of Lesley College has scheduled several Logo courses for the summer session. Harriet Deane, Lesley College, 29 Everett Street, Cambridge, MA 02238. (617) 868-9600 x 360.

A course on advanced Logo will be taught at Long Island University this summer. For details, write Patricia Devlin, Department of Educational Technology, Long Island University, Greenvale, NY 11548. (516) 299-2199.

Looking for a course in advanced Logo for this summer? Contact Dr. Richard Forcier, Educational Technology Program, Western Oregon State College, Monmouth, OR 97361. (503) 838-1220 x 441.

Purdue University has scheduled a course entitled Logo and Elementary Computer Graphics for this summer. Write Dr. James Lehman, Department of Education, Purdue University, West Lafayette, IN 47907. (317) 494-5670.

Teachers College of New York City offers a course called Teaching Logo this summer. For details, contact Howard Budlin, Teachers College, Box 8, 525 West 120th Street, New York, NY 10027.

Courses in Logo for the beginner, the intermediate, and the expert are scheduled this summer at the University of California at Riverside. For more information, write to Sue Teele, University Cottage, University of California at Riverside, Riverside, CA 92521. (714) 787-4361.

The University of Montana offers two Logo courses this summer: Turtle Graphics for Teachers, and Logo in Geometry for Teachers. Contact Johnny Lott, Department of Mathematical Sciences, University of Montana, Missoula, MT 59812. (406) 243-2403.

Logo in Education, a 4 quarter hour course, is offered by the University of Oregon, June 25 - August 16. Contact Fred Beisse, Department of Computer and Information Science, University of Oregon, Eugene, OR 97403. (503) 686-4408.

Gold Disk is featuring special computer offers good until May 31, 1985. They include Wabash SS SD disks, box of 11, \$12.90, Wabash SS DD disks, box of 10, \$14.90, Terapin Logo, \$74.95, Fontrix \$49. Gold Disk, 24 East Pennsylvania Avenue, Bel Air, MD, 21014. (800) 368-2260.

Single sided double density disks for \$1.00 each in any quantity are offered by Blackship Computer Supply. This is about the lowest price we have seen. Also listed are double sided double density disks for \$1.39 each. Please add \$3.00 for shipping and handling. Blackship Computer Supply, PO Box 883362, San Francisco, CA 94188. (415) 550-0512.