

# The National



# Exchange

Volume 2 Number 4

FORWARD 100!

December 1983

## Students Write The Book On Logo

by Lynne Mass

As I read Nancy Sopp's article, "Logo Messups: Process vs. Product," in the October 1983 NLX, I nodded my head in agreement and chuckled, remembering last year when my students and I also proposed to learn Apple Logo together. We began easily, but then reached frustration with the tutorial and decided that we needed an additional manual. Since we could not find one to suit our needs, we decided that maybe the world needed another Logo book, written for children. And why shouldn't we be the ones to write it?

How does a teacher begin to write a computer book with ten year old co-authors? Let me digress a bit.

When our school got its first computer and I taught myself the initial BASIC lessons, I wrote them down for my elementary age students to use in my classroom so they could begin to learn programming, too. They gobbled up the lessons almost as fast as I could produce them! Then came the dreaded day when I knew they were ahead of me ... I couldn't answer their questions!

Having been left in the "computer dust" by my eager students, I decided to work on converting my manual to publishable form as more and more lessons evolved. While I slowly gained some programming knowledge, I frequently reached levels of unrelenting frustration. One day, in utter despair, I bellowed right out loud, "I am so dumb! I can't figure this out!"

Debbie, a fifth grader with a twinkle in her eye, mercifully broke the silence that followed with, "You're not dumb. You're just a slow learner!" We all had a good laugh on that one!

Then Apple Logo arrived! We began investigating this new world, still vaguely discussing the possibility of publishing our own manual. It was not until my BASIC manual was printed that the children really got excited about writing! They organized a format for their manual, decided what to include, and proceed to teach themselves the essence of turtle graphics. After agreeing upon a title, the students developed each lesson in rough form. After rewriting, it was "field tested" in class, and further revised.

After helping to bring it to final form, I contacted Tom Kemnitz of Trillium Press in New York. He sent the students and me contracts to sign, with royalty agreements. It was fun to sign them with the children (and their parents).

Our writing project developed Logo skills, writing and editing skills, and introduced the students to the real world of publishing, publisher's requirements, deadlines, compromises, and legalities. When I asked them what they liked the most, they shouted, "The royalties!"

Whatever it was that motivated them, they were so enthusiastic that they began their second manual on TI Logo ... when summer vacation began! This has just been completed, and will appear in print soon.

As happened to Nancy Sopp, my students and I learned alongside each other, working as colleagues. We learned about Logo and publishing and having fun at the same time!

(Ed. Note: Kids Working with Computers: An Apple Logo Manual for Turtle Graphics is available from Trillium Press, Box 921, Madison Square Station, New York NY 10159.)

Lynne Mass is an elementary science coordinator at Abington Friends School, Jenkintown, PA, and has published a series of books entitled, Kids Working with Computers.

# Guest Editorial

by Steve Tipps

## Pushing the Turtle

Glen Bull and I wrote an article for the special Logo issue of The Computing Teacher (December/January). The article, entitled "Problem Spaces in a Project Oriented Logo Environment," is also printed in serial form in this issue of the NLX and the next. The article is the result of two years of trial and error with Logo. Logo is as exciting as it was two years ago, but our ideas about using Logo are not the same. We have grown and changed in our understanding of Logo.

We were very idealistic in our beginning days, because we felt the benefits of Logo were self evident. Logo was a model for what we believed education should be - a continual process of discovery and collaborative learning. Logo has borne out our beliefs. However, two years have forced us to be more realistic about the world Logo lives in.

While Logo's world is infinitely extensible, our world is not. Logo knows no boundaries of time; we do. Logo is patient and good natured eternally, demanding only that we obey the rules which we give it. We are neither eternally patient, nor infinitely good natured, and only rarely consistent about the rules we like and want. As brilliant as Logo is as a model for learning, it lives in an imperfect human environment.

In our article, we have tried to communicate our view of how Logo might coexist with mortals. The first time we suggested that a project with Logo might be the creation of a microworld, our students (classroom teachers) looked at us as if we were crazed. They retorted that they could not make a DYNATURTLE world, and demanded to know what other microworlds there were. It was the only example we had. We had a bug in our communication and our thought.

The idea of a problem scheme - designed by a teacher for guided discovery and laden with implicit information - has been a thought of Glen's for two years. Playing with NIM in one class convinced us that Logo could be used for programming, problem solving, and learning about number theory.

But, other than ANIMALS, we could not find examples which met our criteria for well developed problem schemes. Assignments to invent problem schemes fell on confused ears.

Next, we suggested Logo projects which were useful personally or professionally. We wanted people to see the potential of Logo to do anything that they could dream up. Challenge the turtle to be used in everyday life, not just to draw squares. This assignment resulted in exciting applications: audiogram plotters, report forms for reading and hearing tests, sports scorekeepers, various INSTANT arrangements, and many cooperative classroom projects. We were beginning to get the hang of describing Logo projects.

"Problem Spaces" is our most recent attempt to suggest Logo projects. Many elements have been organized into four problem spaces which describe different kinds of Logo learning. If the plan does not communicate immediately, we ask that you reread it six months from now. If our framework inspires you or incites you, please let us know. The ideas are worth debating. Debugging the problem spaces seems essential in broadening the niche for Logo. Logo needs more room to survive in our human world.

By looking at Logo in a bigger way, more opportunities for personalizing learning will be available to teachers and students. We want both teachers and students to see their "subjects" in a new way. With Logo, information becomes the byproduct of learning, rather than the end product. The end product of learning should be the desire and the need for more learning.

Finding good Logo examples is still a problem, but we are working on it. We are convinced that books with 2001 preprogrammed Logo projects are not the answer. Good examples are likely to be the result of groups of teachers and students pushing the turtle into new areas. The aim of the NLX is to provide the Logo community with a forum for sharing ideas that work or nearly work. Reading the NLX is half of your involvement in the Logo task. Your ideas and questions are needed for the NLX to fulfill its mission. Consider the NLX an invitation.

FD 100!



# Problem Spaces in a Project-Oriented Logo Environment

by Glen Bull and Steve Tipps

(Ed. Note: This article, published here in two parts, appears in its entirety in the December/January issue of The Computing Teacher. The article may be reprinted provided full credit is given the authors and the NLX, and that a copy of the reprint is sent to the NLX.)

## Part I

A recent review in Consumer Reports characterizes Logo as a language that has gained a wide following among educators as a way to teach programming concepts to children before they have sufficient reading, writing, and math skills to tackle the BASIC language. (See "The Computer as Teacher," Consumer Reports, Vol. 48 (November 1983), p. 617.)

Programming concepts are readily learned through Logo, but Logo is more than an efficient way to introduce programming. Such misunderstandings about Logo are not uncommon. Many introductory texts discuss Logo solely as a "language for children" or as a "way to teach computer literacy." These statements unduly restrict Logo's intent, both as a language and as an educational philosophy.

Using Logo also develops problem-solving skills; the structure of the language encourages a modular approach to problems. Features such as a resident screen editor facilitate a strategy of successive approximations to a solution, since procedures can be altered in seconds.

Logo can also be a vehicle for learning content. Papert presents the imagery of learning math in "Mathland" in the same way that it is possible to learn French through a stay in France. Acquiring programming and problem-solving skills, and developing concepts in subjects such as math and English are important parts of Logo environments.

Of these three, the potential of Logo as a meaningful way of learning content has been neglected. Using Logo for learning subjects ranging from beginning reading to college physics have been imagined, but such applications, though being explored, are not yet in widespread use. Classroom activities which use Logo to explore and develop concepts in subject areas are needed as alternatives to traditional didactic teaching styles. Otherwise, the Logo environment is incomplete.

## PROJECT ORIENTED LEARNING

Consider the following proposition:

"Effective Logo learning is project oriented."

In a project, the content to be learned is embedded in a goal meaningful to the learner. The need for personally satisfying goals can be found in initial use of the REPEAT statement. The first need for repetition can be induced by a desire to avoid repeated retyping of an instruction, but children soon discover great joy in the unpredictable effects of randomly typed commands repeated a large number of times. These seemingly random instructions often produce beautiful and symmetrical designs. Some exploration of this type must be encouraged. However, failure to go on to purposeful activity will produce a row of children gazing in stupefied boredom at 5,000 shifts in the background color of the screen.

To return to Papert's imagery, French is most effectively learned when it is embedded in another context and is not in itself the primary goal. Yet even the presence of the French environment is not enough - Americans in Paris have survived on hamburgers from American hamburger chains.

Subjects such as mathematics and English are also learned best when embedded in a meaningful project. To the student using Logo, the subject matter content learned should simply be something which must be understood in order to do something else which the student wants to accomplish.

Finding projects and applications which address specific content areas seems to be a prime difficulty in implementing Logo based school programs. The problem stems from perceptions of limitations and insufficient illustrations of classroom use rather than

## Problem Spaces continued

the language itself (any memory limitations aside). The relative newness of Logo on widely available microcomputers means that most teachers and students have not pushed the boundaries of Logo beyond turtle graphics. The notion that teacher selection of specific content invalidates the discovery principle is also an inhibiting factor.

Removing these barriers permits a broader view of Logo and makes opportunities for learning content areas with Logo more apparent.

### PROBLEM SPACES

Two dimensions are important in defining how Logo might be used in schools:

- Who provides the idea for the project?
- Who solves the problem and programs the solution?

A two-dimensional matrix defines four possible problem spaces in a Logo environment:

Implementation (Solution)	Source of the Project	
	LEARNER	FACILITATOR
LEARNER	I	III
FACILITATOR	II	IV

#### I. Learner Generated / Learner Implemented Problem Spaces.

In a Type I project, the learner generates both the problem and the solution. This is often considered to be the most natural, "Logo-like" environment.

Even in Type I Problem Spaces, the teacher has considerable control over the content to be learned. If the teacher supplies the set of primitives dealing with turtle graphics, the learner is apt to develop projects which involve geometry, physics and trigonometry. If the teacher supplies primitives which can be used to take apart and combine words and sentences, the learner may embark upon poetry generation and other projects which involve a natural language.

If the teacher supplies tailor made procedures, the learner is apt to explore that particular set of ideas. A BOUNCE procedure modeling reflection of light rays or billiard balls would encourage exploration of vectors.

The underlying structure of Logo encourages strategies such as patterning, modularity, comparison and approximation. At the same time, the learner discovers the possibilities and limitations of the rules which govern the natural microworld of the turtle. This world offers opportunities for learning a new vocabulary, enhancing thinking skills and discovering a consistent rule set.

Ideally, each new project inspires further learning. Each bug, error, or "what if" demands satisfaction and solution. The need for REPEAT begins with pattern recognition. The rationale for procedures is developed when Logo users need an easy way to recall previous events and change them. Implementation of REPEAT statements or of the editor cannot be transmitted telepathically. Providing information about the vocabulary and structure of Logo is the facilitator's responsibility. A facilitator must at time instruct.

The natural learning space of exploration in Logo is not without difficulties for both learners and facilitators. This environment of exploration demands a non-authoritative role for teachers. When a learner asks, "Can I accomplish this?", the answer may be, "How do you think that might be done?" or "Show what you want the turtle to do." These replies are invitations for learners to extend their skills. Some people discover new applications for known skills as they describe their intended goal. Others seem to refuse the challenge of thinking about their own question. They are discouraged because Logo is not automatic ... like a video game.

#### II. Learner Generated / Teacher Facilitated Problem Spaces

In a Type II Logo project, the learner may generate an application which requires skills beyond those possessed. The teacher can choose to direct the learner to an area which is more consonant with that learner's existing level of computing skills. However, there can be value in working out the

## Problem Spaces continued

description (and, if possible, a solution) of the problem in English, even if someone else codes the program. A system analyst, for example, may work out the flowchart or pseudocode for a problem that is coded by a team of programmers. In a TYPE II project, the learner fills the role of system analyst, and the teacher serves as programmer by providing the actual coding.

Beyond the skills required for development of the logic underlying the solution, two other benefits may result from a Type II project. The learner may have the opportunity to observe firsthand the false starts and problems encountered as the teacher attempts to code the problem. When a learner asks how to do something which is not immediately apparent to the teacher, the question may become the basis for a cooperative project. This type of collaboration is not possible when the learner senses that the teacher already knows the answer.

The second benefit is the possible introduction of a new command or concept in a meaningful context. The teacher's version of the program can serve as a model or exemplar which the learner can use to generate solutions to similar problems.

(To be concluded in the January 1984 NLX.)

## NLX ABC's Lead to List Processing Adventures

by Jim Fry

There is life beyond turtle graphics. I found this out while teaching Logo to a summer class of 8 students between the ages of 8 and 13. After a week of turtle graphics, we began using the NLX ABC's to create patterns. Soon, the students were drawing words and sentences on the graphics screen with them.

From there, they showed an interest in printing on the text screen. Thus began our adventures with list processing. After teaching Logo how to print out lists of names and telephone numbers, they began to search for more ambitious projects.

Why not an adventure game? The idea was an instant hit!

(For the benefit of NLX readers who are not familiar with adventure games, here is a brief explanation. A set of rooms or caves or locations is programmed into the computer, based on a map drawn on paper. Connections between the locations are specified so that players can move from one place to certain adjacent others. Events are programmed to happen when players reach certain locations. For example, a monster might attack, or a treasure might beckon. Sheer survival is one important objective of an adventure game!)

We decided to make our adventure take place in a house with four rooms and a main hall. Players could move from room to room by typing compass directions N, S, E, or W.

Here is a sample of what might happen to you. You start at the front door, with the option of entering or chickening out. When you enter, you find yourself in the main hall. Logo asks, "Which direction?" You type N, and find yourself in the ballroom. You spy a monster lurking in the dark corner! The computer asks if you want to fight or flee. If you fight, you must choose a weapon. If you select the right one, you kill the monster. But, if you pick the wrong one, you die, followed by, "Do you want a large or small funeral?"

There are many other surprises waiting in the adjoining rooms. My students had some exciting times planning, writing, and programming the game. They also enjoyed watching others play it. And, they learned a lot about list processing along the way! I was pleased to find the NLX ABC's acted as a natural transition to the list side of Logo.

(Ed. Note: For a free listing of the procedures used for this adventure game, send a self addressed stamped envelope to Jim Fry, 37279 Six Mile Road, Livonia MI 48152.)

Jim Fry teaches math, reading and Logo to elementary school students in Novi, MI, and is the coordinator of LIFT, an organization to help Logo teachers through workshops, consultations, and materials. He is a frequent contributor to the NLX.

# TIPPS for TEACHERS

by  
Steve Tipps

## The Issue of Instant

Instant Logo, or single key Logo, is a set of procedures which moves the turtle around without typing the entire command. For instance, the F key might automatically cause the turtle to move FORWARD 10 and the B key BACK 10. Programs of this kind are available on utility disks, in many Logo publications, including Abelson's Logo for the Apple II and Apple Logo (Byte-McGraw Hill, 1982), and as commercial software such as Delta Drawing by Spinnaker.

Philosophical and mechanical issues of Instant need discussion. Should Instant be used at all? If so, why? When? How? What are the features that make an Instant program good or bad?

After playing with various Instant programs, I have concluded that Instant:

1. is not a substitute for Logo.
2. can serve as a bridge to Logo.
3. allows people access to the power of computers.
4. provides a model for making the keyboard "alive." This has interesting possibilities for many projects in Logo.

I will expand these propositions.

### NO SUBSTITUTE

Instant is typically associated with introductory turtling. The turtle can move around the screen, draw figures, change colors, clear the screen, and even recall and replay its previous moves. This is a subset of beginning Logo features.

Logo is more than turtle graphics. Full Logo must exhibit capabilities in list and numerical operations, and possess a modular structure which encompasses recursion, variability, and interactivity. Finally, and perhaps foremost, Logo must be extensible, that is, it must have the capability to expand through the creation of new commands.

By definition, Instant is not Logo. In my opinion, advertisers' claims that a software package which moves a thing around a screen is "better or easier or cheaper" than Logo are nonsense. You might as well assert that looking at a postcard of the Eiffel Tower is better and easier and cheaper than going to Paris. The two experiences are not equivalent.

Could someone invent a single key Mega-Instant which included all of the features of Logo? Given enough keys on the keyboard for all Logo commands and enough memory to accommodate both the Instant program and the history being generated, a Mega-Instant is possible, but pointless. Instant puts a cocoon around Logo which may obscure the immediacy and flexibility which are part of natural Logo. However, the cocoon can serve a purpose. A well designed Instant serves as a stage in learning a little about Logo.

### BRIDGE

A philosophical question is how to design an Instant so that unintended Logo learning is not lost. I propose a limited definition of Instant turtling as an introduction to Logo. No instant program should do for a child what that child is capable of doing independently. Too much Instant robs the child of the delight of working out problems for themselves.

Teachers in our Logo classes write essays in response to any idea they find interesting from Mindstorms. They write passionately, critically, and perceptively about Logo, teaching, and learning. Two passages by teachers at Thomas Jefferson Elementary School in Falls Church, Virginia, illustrate two approaches in learning to make a square.

Jane Quick wrote:

"Two of my third graders wanted to make a box. They gave the commands FD 100, RT 100, FD 100. At this point they realized that the turtle had turned more than the square corner they wanted. Although just a little uncertain what to do now, they went back and tried RT 80 (too small), then 89 (still too small), then 95 (a little too big), 92, 91, and 90. They knew immediately that they had it! They had discovered for themselves that a square corner was exactly 90 turtle steps.

## Tipps continued

"They were thrilled by their discovery! They had bolstered their self confidence. The next, perhaps more difficult, challenge would be less threatening for them. They had discovered that their initial mistake was not really so terrible, because it had been relatively easy for them to fix their thinking by working together."

On the other hand, Carol Monsess reports:

"I have several students this year who had taken a computer course in Logo this past summer. When they told me, I got the impression that they knew a lot about Logo. They certainly had learned a lot of terms; they could tell me the names of the pieces of equipment and also seemed to know how to make the turtle move and to make squares and various other shapes. I set up the computer for several of them to work on and to show me what they could do.

"After a few minutes, one came and told me there was something wrong with the turtle because it wouldn't make a square. So I went over and everything looked like it was hooked up right and working. I asked them to show me what they had done. The little girl pushed the letters SQ and said, 'See? When I push the SQ, the turtle didn't make a square!'

"This seemed to be a perfect example of the wrong way to teach Logo to children. Given this powerful learning tool and bright children, all the learning had been taken out of Logo. I explained how this turtle had to be told what to do in order to make a square ... Later, the same child told her mother that, if they got a computer, she wanted one that knew more because it was less work. It made me realize that ... even Logo can be taken and taught in such a way that much of the learning is lost."

Instant programs should not take the learning out, but should be a transition for children into Logo. The mechanics might be very different for children of different ages and abilities. The typical Instant turtling program uses keys on the keyboard as shorthand or abbreviations for actual Logo turtle commands.

F key for FORWARD  
R key for RIGHT  
C key for CLEARSCREEN  
etc.

The turtle moves immediately upon touch of key. The programs are available in Harold Abelson's and Dan Watt's books and on Logo utility disks, and should be considered as models for your own Instant procedures rather than fixed programs to be used exactly as they are written. The Instant program in the book may not be the right one for your purposes.

### DISREGARD QWERTY

For some children (ages 3 and 4, and physically limited), the goal of Instant is to remove the keyboard effectively as a barrier to turtling. Control of the turtle is the objective. However, the specific letters may not be important. Design an Instant program and disregard the letters on the keyboard. Keys can be covered with adhesive dots and made into anything you want: arrows, shapes, symbols. The Instant keyboard can be arranged more spatially, or logically than is dictated by the QWERTY layout.

For instance, the RIGHT turn could be controlled by a key on the right side of the keyboard and the LEFT turn being on the left side of the keyboard. Presently, the L is on the right side, and the R is on the left side. So much for teaching any meaningful spatial responses. Perhaps the R key could be used for LEFT turn and U key for RIGHT.

Choosing keys for FORWARD and BACK is complicated because they are easily confused with up and down. Putting them on the same row seems like the best arrangement. Perhaps you might use the F key for FORWARD (coincidentally) and the J key for BACK. My personal design goal is placing keys so as to encourage putting fingers on the home row of the keyboard.

The screen commands such as DRAW, HOME, and CLEARSCREEN might be added on the bottom row keys V, B, and N respectively. The basic Instant keys would be arranged in the center of the keyboard.

With those few Instant keys, turtling is possible. Embellishments are infinite. The teacher can respond and add commands as the children seem to grow to them. HIDE-TURTLE and SHOWTURTLE could be added at the center of the keyboard on H and G. Pen colors and background colors might be added on the top row of the keyboard by sharing the same number keys. Background colors could be reached using the shift key. Shapes



## more Tipps

such as squares, triangles, stars, and circles could also be programmed and made available through other keys.

The sizes of the turtle movements and shapes should be scaled so they can be used together. After experimenting with different sizes, I think that movements of 20 or 30 turtle steps and 30 turtle turns are a good starting place. Smaller sizes don't seem to have enough effect to be really interesting.

Whether you add colors or shapes or anything else is your choice. The first law of Instant is to suit the procedures to the level and needs of the children without denying them the joy of discovery. For a four year old, having an Instant square may inspire creative squaring. For a seven year old, an Instant square may interfere with finding RT 90. For a cerebral palsied 30 year old, Instant square may be the first opportunity to demonstrate spatial understanding and creativity.

### MISTAKES I HAVE MADE WITH INSTANT

I have designed several Instant procedures and made numerous mistakes. For three and four year olds at the University of Virginia Child Development Center, I decided on a turtle control system so that the turtle moves were like points on a compass. Cursor control on Wordstar uses a logical key arrangement with E for up, X for down, D for right, and S for left. I adopted this spatial arrangement for an Instant program with four additional keys going northeast, southeast, southwest, and northwest.

Initially, I programmed it so that, with a single keystroke, the turtle turned and moved simultaneously. The resulting confusion over distance and direction probably interfered with the learning of turtle control. This confusion is apparent to me now; I think that separate keys for distance and direction should be used. For moves, 10 as a unit seemed too small for my audience; the square was barely discernible from the circle. I am still debating whether programming Instant squares, circles, and triangles is good for those children. I know they use them and make wonderful designs and interesting patterns, but wonder if they would do the same with only the turtle movements.

## PROCESS AND PRODUCT

Other issues in Instant are whether a history of all the moves should be generated, whether procedures should be definable through Instant, and whether saving work should be built into Instant. If the objective for the children is control of the turtle, I see no reason for these being provided.

It must be recognized that this moves away from the process of Logo and toward a product. However, some students may need those single key abilities. Having an easily reproducible record can facilitate project orientation and problem solving. Procedures which save a history and allow the product to be redrawn or to undo the last move are probably very helpful. If the history was defined as a procedure, however, I would insist that the procedure be named meaningfully. Using numerals as procedure names (as in Delta Drawing) is counterproductive to Logo.

Five and six year olds often want to keep their work; Instant keys for READPICT and SAVEPICT might be written. To be consistent with Logo, the Instant program should ask for a name of the picture for labeling the picture on the disk. Even if only personal names were used, meaningful names are maintained.

### INSTANT FOR UNDERSTANDING

Instant programs assume that the user will recognize the three part relationship of F, FORWARD, and forward turtle movement. At least two or three variations of Instant might be used which do not make this assumption about the learners.

Tom Lough's Slow Turtle procedures (NLX September 1983 and Teaching and Computers magazine, September 1983) could be built into the Instant program. When F is pushed, the turtle could move FORWARD 20 or 30 spaces, one space at a time. Another way to change Instant to emphasize what is going on is to have the computer print FORWARD 30 when F is pushed. These two additions are very simple variations to a standard Instant program.

TO START  
DRAW  
INSTANT  
END



## Tips . . .

```

TO INSTANT
  TEST RC?
  IFTRUE ACTION
  INSTANT
  END
TO ACTION
  MAKE "KEY RC
  IF :KEY = "F PRINT [FORWARD 20]
  REPEAT 20 [FORWARD 1]
  IF :KEY = "B PRINT [BACK 20]
  REPEAT 20 [BACK 1]
  IF :KEY = "R PRINT [RIGHT 15]
  REPEAT 15 [RIGHT 1]
  IF :KEY = "L PRINT [LEFT 15]
  REPEAT 15 [LEFT 1]
  END

```

The number of keys which are "alive" can be expanded to include any other needed commands. The size of the moves and turns can be changed. The point is to have a program which is explicit in what is happening when the key is pressed.

Another halfway place to consider is to have an Instant program which requires a return key to be pressed to execute the line. CHAR 13 is the ASCII code for RETURN. ACTION is changed so that after the command is printed, RETURN must be pressed to get the turtle to move.

```

TO ACTION
  MAKE "KEY RC
  IF :KEY = "F PRINT1 [FORWARD 20] F
  IF :KEY = "B PRINT1 [BACK 20] B
  IF :KEY = "R PRINT1 [RIGHT 15] R
  IF :KEY = "L PRINT1 [LEFT 15] L
  END
TO F
  IF RC = CHAR 13 REPEAT 20 [FD 1]
  PRINT CHAR 13 STOP
  F
  END
TO B
  IF RC = CHAR 13 REPEAT 20 [BK 1]
  PRINT CHAR 13 STOP
  B
  END
TO R
  IF RC = CHAR 13 REPEAT 15 [RT 1]
  PRINT CHAR 13 STOP
  R
  END
TO L
  IF RC = CHAR 13 REPEAT 15 [LT 1]
  PRINT CHAR 13 STOP
  L
  END

```

Another bridging idea comes from the British version of Instant Logo. The function keys on the ACORN microcomputer are programmed to print the commands FORWARD, BACK, RIGHT, and LEFT to which the input number is added from the keyboard directly. You gain the efficiency of typing without predetermining the size of the movements.

All of the versions are intended to make the connection between pressing a key and the result. None are substitutes for real Logo and many people will find them unnecessary diversions. You must make the judgement.

## INSTANT ACCESS

Instant can remove the difficulty of using the keyboard for interacting with the computer and the turtle. Instant provides access to the computer to anyone who has difficulty with keyboard use. Two groups are immediately evident: young children and people with motor difficulty. Other special populations (autistic, EMR, and the like) might also use Instant's immediate interaction with the computer environment. The idea would be to provide a computer environment which is automatically responsive.

Alternate input devices, such as electronic pens, touch sensitive screens, and voice recognition systems also allow control of the computer. Instant can be considered to be a software appliance. For a teacher or worker with special populations, the possibilities of tailoring the keyboard to the particular learning needs are infinite.

However, Instant turtling is only one possibility in providing expanded access to computer functions. Instant drawing pads with a choice of designs (Koala Pad or Power Pad, for example), Instant message producers with a varied menu of messages, Instant calculators with demonstrations of the actual operation -- all kinds of Instant programs can be made for people who need the power of the computer to do interesting or important things for themselves.

Instant programs allow all of us access to the computer's power. These applications do not "teach." They merely are enabling environments. Word processing and accounting software might also be considered Instant programs which bring the power of those functions at the press of a key. Glen Bull and I have always contended that word processing and spreadsheet programs on the computer were Logo-like because they impart power to the user. Our definitions of Instant have been much too limited.

## Tipps . .

### INSTANT PROJECTS

All of these Instant ideas provide many projects for students and teachers. The keyboard can be transformed into any function. A key scramble (in which students change the letter printed in response to pressing a key) is a fun introduction to the Instant keyboard. Personal coders and encoders can also be designed. The attitude of Logo is, "I can make this computer do anything or become anything." Instant is an imperfect extension of Logo philosophy.

None of these projects are Logo. The same programs and appliances could be and have been written in other languages. The power of Logo is its adaptability to any use which can be dreamed up. Logo allows an everyday person (parent, student, teacher, business person) with a knowledge of Logo to create a responsive environment of any type.

(Ed. Note: For information on drawing pads which can be used with Logo, contact Koala Technologies Corporation, 3100 Patrick Henry Drive, Santa Clara, CA 95050, and Chalkboard, 3772 Pleasantdale Road, Atlanta, GA 30340.)

---

Steve Tipps is a professor in the University of Virginia's School of Education, and presents Logo workshops for school systems throughout the United States.

---

## Chicago LLX News

Andy David has started a local Logo exchange for persons in the Chicago area. The first meeting was held October 15, and was attended by over twenty Logo teachers, teacher trainers, programmers, publishers, and consultants. Among the many ideas under consideration by the group are the establishment of a software exchange, test and evaluation site, and the regular publication of a many-to-many newsletter. Each newsletter subscriber sends in a short article of an agreed-upon length. The editor compiles all articles, prints them, and redistributes the copies to the contributors. For additional information, write Andy David, 5430 South Harper Avenue, Chicago IL 60615, or call (312) 684-3178.

# MICROWORLDS

by  
Glen Bull

## Logo Grade Book

### Part II Report Generator

In the last column, I discussed a method for using Logo to record and average grades. This month, I will review the use of list processing to generate a report of the recorded grades.

The form of the report ideally should resemble a page from a gradebook:

	#1	#2	#3	#4	AVG
John	89	94	ABS	91	91
Mary	88	ABS	79	86	84
Sam	87	91	93	88	90
Sally	94	95	92	100	95

#### ENTERING NAMES AND SCORES

In addition to the test scores for each student, a list of the names should be entered into the Logo gradebook:

MAKE "CLASS [JOHN MARY SAM SALLY]

MAKE "JOHN'S.Scores [89 94 ABS 91]

MAKE "MARY'S.Scores [88 ABS 79 86]

MAKE "SAM'S.Scores [87 91 93 88]

MAKE "SALLY'S.Scores [94 95 92 100]

The class names and scores can be stored in a file as described in last month's column.

#### STARTING THE REPORT

The steps involved in printing the grades of a list of students involves the following:

- Print the name of the first student in the list
- Print the test scores of the student
- Print the student's average
- If there are any more students, go to the next one, otherwise stop.

The basic structure outlined can be tested with a short procedure to print all the names in the list of students.

## Microworlds continued

```
TO PRINT.LOOP :CLASS
  IF :CLASS = [] STOP
  PRINT1 FIRST :CLASS
  PRINT.LOOP BUTFIRST :CLASS
END
```

This procedure works after a fashion, although the result seems odd because there are no spaces between the names:

```
PRINT.LOOP :CLASS
JOHNMARYSAMSALLY
```

Nevertheless, the procedure does print all the names of the class, and then stop as we asked.

The next task is to ask the procedure to print the student's scores as well. In the immediate mode, the scores of the first student could be printed by typing:

```
PRINT :JOHN'S.Scores
89 94 ABS 91
```

In order to make the procedure print the student's scores, it is necessary to ask the procedure to create a new word consisting of the student's name JOHN and the suffix 'S.Scores, to form JOHN'S.Scores. This task can be accomplished with list operator WORD, a built-in command in Logo. The command WORD is used to put letters together to form words. Before going on, it may be desirable to recall that:

- A list is a series of items enclosed between brackets. [ JOHN MARY SAM SALLY ]
- A word is an unbroken series of letters preceded by a quotation mark. "AVOCADO

Two words can be joined together to form a single word:

```
PRINT WORD "FIRE "TRUCK
FIRETRUCK
PRINT WORD "JOHN "S.Scores
JOHNS.Scores
```

This is close, but we still need the apostrophe.

### PRINTING APOSTROPHES

The apostrophe is a reserved symbol in the MIT versions of Logo (Terrapin and Krell), and will not print directly because its use is reserved for other purposes.

The apostrophe is needed here to form the word JOHN'S.Scores. Fortunately, each character has a number associated with it. Even when the character can not be printed directly, it can be printed by typing the Logo command CHAR in front of its number. For example, the number associated with the character A is 65, and the number associated with an apostrophe is 39.

```
PRINT CHAR 65
A
PRINT CHAR 66
B
PRINT CHAR 39
'
```

This approach can be used to create a word which will be used to access John's scores:

```
PRINT ( WORD "JOHN CHAR 39 "S.Scores )
JOHN'S.Scores
```

The parentheses are essential. The command WORD normally expects only two inputs. The parentheses override this expectation so that three or more inputs can be combined to form a single word.

There is one other point to note. In many cases, spacing around parentheses is not crucial in Logo. Here, however, there must be a space between "S.Scores and the final parenthesis. Otherwise, Logo will think that the parenthesis is part of the final input.

### THINGS AND NAMES

Now that the word "JOHN'S.Scores can be formed, it is possible to access the THING associated with "JOHN'S.Scores. THING is another Logo command which is the equivalent of the colon used to access the thing associated with a name. The following two commands are equivalent in Logo:

```
PRINT :JOHN'S.Scores
89 94 ABS 91
PRINT THING "JOHN'S.Scores
89 94 ABS 91
```

Since this is the case, THING can be used to access data associated with a name created through use of the list operation WORD.

```
PRINT THING ( WORD "JOHN CHAR 39 "S.Scores )
89 94 ABS 91
```

## more Microworlds

This equivalent instruction is starting to have a cryptic appearance. It says to print the thing associated with word formed by combining "JOHN, and apostrophe (CHAR 39), and "S.Scores. Another step is required before the instruction can be inserted into the procedure PRINT.LOOP. In PRINT.LOOP the name "JOHN is not accessed by typing the name directly. It is obtained by referring to the first item in a list of class names.

```
PRINT :CLASS
JOHN MARY SAM SALLY
PRINT FIRST :CLASS
JOHN
```

Therefore, typing FIRST :CLASS is the equivalent of typing "JOHN directly. This is shown below:

```
PRINT THING ( WORD "JOHN CHAR 39 "S.Scores )
89 94 ABS 91
PRINT THING ( WORD FIRST :CLASS CHAR 39
"S.Scores )
89 94 ABS 91
```

This new instruction, which is the equivalent of PRINT:JOHN'S.Scores, can be inserted into PRINT.LOOP.

```
TO PRINT.LOOP :CLASS
IF :CLASS = [] STOP
PRINT1 FIRST :CLASS
PRINT THING ( WORD FIRST :CLASS CHAR 39
"S.Scores )
PRINT.LOOP BUTFIRST :CLASS
END
```

### SPACING THINGS OUT

The revised procedure does print each student's name and test scores, but the formatting leaves something to be desired:

```
PRINT.LOOP :CLASS
JOHN89 94 ABS 91
MARY88 ABS 79 86
SAM87 91 93 88
SALLY94 92 95 100
```

One difficulty is that there is no space between each student's name and their first test score. CHAR 32 is a space, just as CHAR 39 is an apostrophe and CHAR 65 is an A. Therefore, printing CHAR 32 will produce a space in between the name and the test score. Although this instruction could be inserted directly into PRINT.LOOP, the creation of a procedure, SPACE, to print a variable number of spaces would be both more general and less cryptic.

```
TO PRINT.LOOP :CLASS
IF :CLASS = [] STOP
PRINT1 FIRST :CLASS
SPACE 1
PRINT THING ( WORD FIRST :CLASS CHAR 39
"S.Scores )
PRINT.LOOP BUTFIRST :CLASS
END
```

```
TO SPACE :NUMBER
REPEAT :NUMBER [PRINT1 CHAR 32]
END
```

Inclusion of a space between the student's name and the test score helps the appearance of the format somewhat:

```
PRINT.LOOP :CLASS
JOHN 89 94 ABS 91
MARY 88 ABS 79 86
SAM 87 91 93 88
SALLY 94 92 95 100
```

### LINING UP NAMES

It is now apparent that one of the reasons that the scores do not line up is the different lengths of student's names. One solution to this problem would be to require the students to acquire names of the same length. This solution was tried in the early days of computing, and met with some resistance!

Another solution would be to count the number of letters in each name, and insert enough spaces before each name to align the right edge of each word with the one above. For example, SAM needs two spaces before to ensure that the last letter in the name lines up with the final letter in SALLY. JOHN and MARY each need one space before them to line up with SALLY on the right-hand side. This type of alignment is called "right-hand justification," and is used in many publications, including the NLX.

A procedure called COUNT.LOOP can be created to count the number of letters in each name. COUNT.LOOP takes a word as an input, and uses BUTFIRST to remove letters from the word, one at a time. As each letter is removed from the word, a variable named :NUMBER is increased by one. When all the letters have been exhausted, COUNT.LOOP outputs the number of letters in the name. As a matter of convenience, a calling procedure LETTER.COUNT supplies a starting value of 0 for the input :NUMBER in COUNT.LOOP each time.

## **Microworlds . . .**

```

TO LETTER.COUNT :WORD
  COUNT.LOOP :WORD 0
  END
  TO COUNT.LOOP :WORD :NUMBER
    IF :WORD = "  OUTPUT :NUMBER
    OUTPUT COUNT.LOOP (BUTFIRST :WORD)
    (:NUMBER + 1)
  END

```

The parentheses in the last line of COUNT.LOOP are only inserted for clarity, and can be removed without affecting the procedure. If the procedure LETTER.COUNT is asked to count the number of letters in "JOHN, it supplies an answer of 4.

```

PRINT LETTER.COUNT "JOHN
4

```

It is now possible to insert a line into PRINT.LOOP which will print six spaces, minus the number of letters in each name. For example, six spaces minus the four letters in "JOHN would result in the insertion of two spaces before the name. In this case, it is necessary to remember that FIRST:CLASS refers to the first name in the list of class names [ JOHN MARY SAM SALLY ].

```

TO PRINT.LOOP :CLASS
  IF :CLASS = [] STOP
  SPACE 6 - LETTER.COUNT FIRST :CLASS
  PRINT1 FIRST :CLASS
  SPACE 1
  PRINT THING ( WORD FIRST :CLASS CHAR 39
"S.Scores )
  PRINT.LOOP BUTFIRST :CLASS
  END

```

### **LINING UP SCORES**

This refinement produces additional improvement in the appearance of the class scores when PRINT.LOOP is run. The same effect could have been produced with the cursor positioning commands in Logo. However, the cursor positioning commands only affect the screen, and do not format the text when it is sent to the printer. Therefore, the use of spaces to format the names will produce a more universal result which will work for both the screen and the printer.

```

PRINT.LOOP :CLASS
  JOHN 89 94 ABS 91
  MARY 88 ABS 79 86
  SAM 87 91 93 88
  SALLY 94 92 95 100

```

Now the names are all right justified, but the test scores are still not aligned. The reason for this is that the scores are not all the same width. Most are two characters wide, but a few are three characters wide. Although we easily could have used AB instead of ABS to indicate an absence, it would hardly be reasonable to ask Sally to accept a 99 instead of 100 on her last examination just to improve the formatting of the scores!

An alternate solution would be to insert spaces before some of the scores to line them up, just as we did with the students' names.

```

TO PRINT.Scores :Scores
  IF :Scores = [] SPACE 1 STOP
  SPACE 5 - LETTER.COUNT FIRST :Scores
  PRINT1 FIRST :Scores
  PRINT.Scores BUTFIRST :Scores
  END

```

This procedure will insert three spaces before each of John's test scores, except for the third, for which John was absent. Only two spaces are inserted before the third test, to maintain an even alignment.

```

PRINT.Scores :JOHN'S.Scores
  89 94 ABS 91

```

The PRINT.Scores procedure can now be inserted into PRINT.LOOP.

```

TO PRINT.LOOP :CLASS
  IF :CLASS = [] STOP
  SPACE 6 - LETTER.COUNT FIRST :CLASS
  PRINT1 FIRST :CLASS
  PRINT.Scores THING ( WORD FIRST :CLASS
CHAR 39 "S.Scores )
  PRINT []
  PRINT.LOOP BUTFIRST :CLASS
  END

```

This revision takes care of the remaining formatting details in PRINT.LOOP. The names and scores are now aligned when the procedure PRINT.LOOP is run.

```

PRINT.LOOP :CLASS
  JOHN 89 94 ABS 91
  MARY 88 ABS 79 86
  SAM 87 91 93 88
  SALLY 94 92 95 100

```

### **AVERAGING THE SCORES**

The remaining task is to print the averages as well as the scores. This can be accomplished with the procedure to average lists of numbers developed in last month's column.

## Microworlds . .

```
PRINT AVERAGE :JOHN'S.Scores
91.333
```

In the formatted grade listing, an average rounded to the nearest integer is desirable. This can be accomplished with the Logo primitive ROUND.

```
PRINT ROUND AVERAGE :JOHN'S.Scores
91
```

This method of printing the average can be inserted directly into PRINT.LOOP.

```
TO PRINT.LOOP :CLASS
  IF :CLASS = [] STOP
  SPACE 6 - LETTER.COUNT FIRST :CLASS
  PRINT1 FIRST :CLASS
  PRINT.Scores THING ( WORD FIRST :CLASS
CHAR 39 "S.Scores )
  PRINT ROUND AVERAGE THING ( WORD FIRST
:CLASS CHAR 39 "S.Scores )
  PRINT.LOOP BUTFIRST :CLASS
END
```

Do not feel that this is the ultimate version. In fact, there are many improvements which can be made. However, I hope that you have gained a fuller understanding and appreciation of the power and capabilities of the word, list, and arithmetic sides of Logo in the process.

### LCSI VERSION

Some of these procedures for LCSI Logo do not seem to work unless they are defined in the editor using EDIT, rather than TO. I am at a loss to explain this, but, until the reason is determined, I suggest that you use EDIT.

Note that, in order for PRINT.LOOP to work, the procedures AVERAGE.LOOP and AVERAGE from last month's column must be included.

```
TO PRINT.LOOP :CLASS
  IF EMPTY :CLASS [STOP]
  SPACE 6 - LETTER.COUNT FIRST :CLASS
  TYPE FIRST :CLASS
  PRINT.Scores THING WORD FIRST :CLASS
  "'S.Scores
  SPACE 3
  PRINT ROUND AVERAGE THING WORD FIRST
:CLASS "'S.Scores
  PRINT.LOOP BUTFIRST :CLASS
END
```

```
TO PRINT.Scores :Scores
  IF EMPTY :Scores [SPACE 1 STOP]
  SPACE 5 - LETTER.COUNT FIRST :Scores
  TYPE FIRST :Scores
  PRINT.Scores BUTFIRST :Scores
END

TO SPACE :NUMBER
  REPEAT :NUMBER [TYPE CHAR 32]
END

TO LETTER.COUNT :WORD
  OUTPUT COUNT.LOOP WORD 0
END

TO COUNT.LOOP :WORD :NUMBER
  IF EMPTY :WORD [OUTPUT :NUMBER]
  OUTPUT COUNT.LOOP BUTFIRST :WORD
:NUMBER + 1
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.

---

## NOVA's "Talking Turtle" — A Compelling Documentary

Reviewed by Andrew David

"Talking Turtle" was the subject of the weekly science program NOVA, broadcast on October 25th on public television stations nationwide. For many, I'm sure, this sensitive documentary furthered the exciting exploration into Logo's larger educational context.

Many of us began this exploration by reading Seymour Papert's book, Mindstorms: Children, Computers, and Powerful Ideas (BASIC Books, 1980). Here, in his eloquent treatment of the pressing subject of computers and education, Papert wrote of an educationally rich culture in which individuals may take control over their own learning. His vision was both hopeful and challenging.

Since then, the task befalling parents, teachers, and children has been a less attractive one of implementation. When do we use the quotes and when do we use the colon? Which version will best serve our purpose? What about lists? These considerations have, in part, distracted us from the forceful social concerns that have motivated Papert and others of the MIT team that developed Logo.

## NOVA continued

Writer Suzanne Campbell-Jones and a competent production team have captured the power of Logo through a series of profiles highlighting its diverse applications. Among those subjects examined were the original floor turtle robot (from which the familiar screen turtle has descended); disabled students who, through Logo, are experiencing a rare opportunity to explore and manipulate space; the Lamplighter Project of Dallas, Texas, boasting America's highest computer to student ratio; an inner city project, where otherwise "non-exceptional" students excel on the computer; and several bilingual versions, including one in Wolof, the native language of Senegal.

These scenes of very real application have been interwoven with pertinent interviews, and, of course, the ideas and the urgings of Seymour Papert. He is as articulate a speaker as he is a writer; his passionate manner and constant use of non-computer metaphors remind us that his focus on computing machines has done little to dilute his original emphasis on people and their needs.

"Education," he tells us, "is much more related to love than to logic. Education has very little to do with explanation - it has to do with engagement." Every moment of the film supports this. A child who can barely control the movement of his hands takes three carefully calculated actions and causes the floor turtle to move a tower of blocks; a young but sophisticated programmer watches her program "fail," then immediately tracks down the problem and corrects it; native Senegalese children program the turtle to draw pictures of huts ... their huts, images out of their own lives.

This is welcome testimony to the variety of experience that is reflected in Logo programming! With equal sensitivity, the filmmakers have shown that Logo's problem solving approach is part of the living, breathing world outside the computer screen as well. Scenes of "turtle play" in a pile of leaves, young dancers spinning on the floor to choreography that has been scribed in a "procedural" format, gears and jugglers (two of Papert's most important metaphors) have all been delicately chosen to enhance the assertions that Logo is not about binary addition and cathode ray tubes, but about living and learning.

"Talking Turtle" was presented in the true spirit of Seymour Papert and his vision. Of course, it is possible to use Logo in a very different way than that which he envisions. But to deprive Logo education of the insights and social concerns from which it was spawned - insights illuminated by such works as Mindstorms and now this NOVA feature - is to insure that its potential will never be realized. As teachers and programmers, we are all struggling to determine how to use Logo. This outstanding production reminds us why.

Andrew David is doing Logo networking and advanced Logo applications research in Chicago. He also works with elementary students citywide in weekly workshops on Art and Perception.

## NLXual Challenges

with Robs Muir

### Starring Logo

If you have spent some time reading Logo manuals and instructional materials, you have played around with POLY. POLY is very good for drawing regular polygons. All regular polygons have one important characteristic; when the turtle finally returns to its starting position, it has turned 360 degrees (no more and no less). One of the interesting aspects of POLY is that you can generate not only regular closed polygons (e.g., square, triangle, hexagon, etc.), but you can also create complex polygons.

A complex polygon is a closed figure (the turtle returns to its starting position and heading) with a total turning of some integer multiple of 360 degrees. Most of us are familiar with a Logo "standard," the five pointed star. However, we usually introduce our students to the complex polygonal form of a star created by

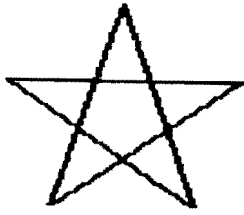
REPEAT 5 [FD 50 RT 144]

This clearly illustrates a total turning of a multiple of 360 degrees ( $5 * 144 = 720 = 360 * 2$ ).

This month's problem is built around such a five pointed simple polygon. Our first challenge is to build a star in which no one side crosses another; the total turning is exactly 360 degrees.



## Challenges continued



Just to make things a bit more difficult, have the turtle start and finish in the exact center of the star. Then, write a procedure which will draw the star of any determined size with a single input.



Once this is accomplished, you are ready for the remainder of the challenge. Draw a five pointed star, and then "stamp" a second star, using the same axis of rotational symmetry. Try different sizes, and different angles of rotation also. Can you make a space filling pattern out of the star and other geometric shapes?

The ancient Greek mathematicians believed that the simple star embodied the deepest knowledge of geometry. Contained within the star is the "golden mean," the basis of much Greek art and architecture, and the pentagon, the mystic symbol of the Middle Ages. Our double star image is also a commonly recurring theme in Islamic art; its complexity and beauty is revered in Middle Eastern decorative art.

The underlying difficulty of this problem may not be readily apparent when you first begin. If you have really plumbed the depths, you should be able to pack the double stars anywhere on the screen in any size. If you have some initial difficulty, remember that humans have studied this figure for at least 25 centuries.

As with all NLXual Challenges, if you want to receive a possible solution, send a copy of your solution (or your attempt) with a letter size self addressed stamped envelope to NLXual Challenges, Attn: Robs Muir, 1688 Denver Avenue, Claremont CA 91711. We will compile contributions and send you the most elegant solution.

If you would like to submit a challenge or puzzle for this column, send your ideas to the above address.

---

Robs Muir is a teacher in the Claremont CA Unified School District and Claremont Graduate School. He presents Logo workshops in the southern California area.

---

## Logo Publication Announced

TurtleTalk is a quarterly newsletter published by Larry Lohrman, of Issaquah, WA. Featured are articles about Logo teaching and learning environments, reviews of Logo versions, explanations of procedures, and examinations of Logo related subjects, such as artificial intelligence and robotics. Larry offers TurtleTalk subscribers a 10% discount on popular Logo books. Subscriptions to TurtleTalk cost \$15 per year (4 issues) for the USA, Canada, and Mexico, \$20 per year elsewhere. For a free sample copy, write: TurtleTalk, 955 Greenwood Boulevard, Issaquah, WA 98027.

The National Logo Exchange, copyright 1983 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 01970. Address other correspondence to: The National Logo Exchange, Box 5341, Charlottesville, VA 22905. ISSN 0734-1717

Editor.....Tom Lough

Need a Logo idea for use before Christmas vacation? Students of Nancy Kovatch, a teacher at St. Anne's - Belfield School in Charlottesville, VA, are designing Christmas cards using Logo! To make a card, print an appropriate Logo drawing on half of a page of paper, and a text message on the other half. The drawing is colored and the paper folded to finish the card. The cards will be special presents for parents, teachers, and friends.

[illegible]

```

TO PR.IT
  MAKE "INPUT READLIST
  .PRINTER 1
  REPEAT 6 [PRINT []]
  MAKE .GREETING
  .PRINTER 0
END

```

Do you know of a source for interesting and well made turtle paraphernalia? Turtle jewelry, toys, puppets, plaques, cards, and the like are all around, and make excellent Logo classroom additions. If you have the name and address of a person or firm which makes high quality turtle items, send us information about them to share with other NLX readers.

On page 31 of the November 14 issue of Infoworld, CompuSource, a company in Minneapolis, announced that its Abacus computer will be both portable and Apple compatible. If you have ever wanted Logo-to-go, this may be it! Boasting 64 K of memory in a 28 pound package, the portable will sell for \$1595.

On page 109 of Infoworld's November 21 issue, Gary Kildall (author of CP/M and DR Logo) reported that Digital Research has shipped 4000 copies of DR Logo in four weeks! He further stated that Sony has an 8-bit version of it in Japan. Digital plans to release a generic 8-bit Logo version sometime in the future. They also plan to release soon a Logo version which will operate with the PCjr.

In the same issue, an interesting item appears on page 112, indicating that an IBM compatible TRS-80 is scheduled for release by Radio Shack in the spring. Since there will be at least four Logo versions for the IBM PC, let's hope that one or more of them will run on this TRS-80 model.

The November 28 issue of Infoworld carried a story about Seymour Papert on page 31, in which he suggested that recent FBI raids on teenaged computer hackers were not good solutions to the problem of computer break-ins. He cited three styles of programming (logical, interactive, and machine-identified), and linked the latter style with computer hackers. Teaching strategies which develop respect for computer systems and for the rights of others to privacy might be effective preventatives for future problems, he noted.

Rumor has it that all IBM PC Logo versions will run on the Compaq and the Columbia personal computers without modification. If you have Logo information on this or any other IBM PC compatible computers, please let us know so we can share your findings with other NLX readers.

Eadie Adamson of the Nightingale - Bamford School in New York City reports that the NLX ABC's work well with her Commodore Logo. The only change she had to make was to add PRINT [ PRESS SPLITSCREEN KEY - F3 - TO BEGIN ] at the end of the WRITE procedure. For a free listing of the NLX ABC's, send a self addressed stamped envelope to NLX ABCs, PO Box 5341, Charlottesville, VA 22905.

LIFT (Logo Information For Teachers & Trainers) is an organization to assist teachers and teacher trainers in implementing Logo in classrooms in Michigan and the midwest. Workshops, consultations, and teaching materials are available. Contact Jim Fry, LIFT, 37279 Six Mile Road, Livonia, MI 48152, (313) 464-8088.

Have you sent for information about the Apple Computer Clubs yet? If not, mail your inquiry to Dick Casabonne, Apple Computer Clubs, PO Box 948, Lowell, MA 01853. They have a great program to aid in setting up computer clubs in schools. You will find several Logo mentions in their organizing materials. Ideas from the May 1983 NLX are featured on the large colorful club poster!

By the way, while you are writing to the Apple Computer Clubs, ask for information about their Computer Club Competition. Here is an opportunity for students to win up to \$1000 for their computer projects. There are categories for elementary and secondary students with both individual and group projects. There is plenty of Logo potential here!

In the November 1983 issue of Softalk, the following line appears on page 336 with no further commentary.

"Apple Logo continues to outsell its competitors by a wide majority."

If you are interested in starting a local Logo exchange, the NLX has available free a set of suggestions and guidelines. Send a self addressed stamped (37c) business size envelope to The National Logo Exchange, Attn: LLX Guide, Box 5341, Charlottesville VA 22905.

What Logo ideas did you use in your classrooms last Valentine's Day? Why not share them with other NLX readers to use this year? Please send a summary of your activities to NLX Valentines, PO Box 5341, Charlottesville, VA 22905, before January 1, 1984.