THE EFFECT OF LOGO (TURTLE GRAPHICS) ON THE PROBLEM SOLVING STRATEGIES USED BY FOURTH GRADE CHILDREN

by

Honi Joyce Bamberger

Dissertation Submitted to the Faculty of the Graduate School of the University of Maryland in partial fulfillment of the requirements for the degree of Doctor of Philosophy

1984

copy 1

Maryland
LD
3231
.M70d
Bam-
berger,
H.J.
FOLIO

© 1984 Honi Joyce Bamberger

APPROVAL SHEET

Title of Dissertation: The Effect of Logo (Turtle Graphics) on the Problem Solving Strategies Used by Fourth Grade Children

Name of Candidate: Honi Joyce Bamberger

Dissertation and Abstract Approved: Martin L. Johnson

Dr. Martin L. Johnson,
Associate Professor,
Education
Department of Curriculum
and Instruction

Date Approved: 11/29/84

ABSTRACT

Title of Dissertation: Effects of Logo (Turtle Graphics) on the Problem Solving Strategies Used by Fourth Grade Children

Honi Joyce Bamberger, Doctor of Philosophy, 1984

Dissertation directed by: Dr. Martin L. Johnson
Associate Professor of Education
Department of Curriculum and Instruction

The major purpose of this study was to determine the effect of learning the programming language Logo on the problem solving strategies used by fourth grade children while they solved mathematical word problems without a computer. A secondary purpose was to investigate whether children who had learned to program with Logo would regard mathematics more favorably as measured by a mathematics attitude survey.

This eleven-week study was designed to teach Terrapin Logo (Turtle Graphics) to 15 out of a total of 30 fourth grade children. The study compared the

audiotaped and written responses to a four-question problem solving posttest to determine which strategies were used by each group and whether the Logo group applied the strategies that they had been taught during their programming experience when solving non-computer problems. The Logo curriculum, written by the investigator, focused on teaching children the basic commands of the language while stressing the need to plan a procedure before beginning it, breaking a large idea into more manageable parts, guessing and then checking work and finally checking over the finished product. Some group instruction occurred, but individual or paired instruction was emphasized. Guided discovery ensured that strategies were being used and commands correctly applied. The statistical analysis consisted of computing means and standard deviations on individuals' attitude responses and a t-test was performed on these means. Chi-square critical values were computed for the process coding sequence, as well as t values.

The following results were obtained from the study:

1. On the attitude measure significant differences were found only between the experimental and control groups on their general attidude toward school.

2. Statistically significant differences occurred on several posttest problems using the strategies of checking and looking back, as well as making an error and then immediately correcting it.

The investigator concluded that the strategies of checking and looking back over one's work and correcting an error were transferred by the experimental group when solving noncomputer word problems. Several important factors may have contributed to the lack of significant differences elsewhere. Among them are 1) short duration of the study, 2) confusion over application of computer strategies to mathematical problems, 3) limitations of process coding sequence, and 4) population size.

ACKNOWLEDGMENTS

I wish to thank the many people who have so generously provided me with assistance, encouragement and support throughout this study.

My deep appreciation goes to Dr. Martin L. Johnson, my advisor, for his continual support, enthusiasm, time, guidance and constructive criticism.

I am extremely grateful to Dr. Patricia F. Campbell for reviewing and editing this dissertation, and for her guidance during the year that she acted as my advisor.

I appreciate the time and assistance given by Dr. Charles Johnson, Dr. James T. Fey, and Dr. Betty Simms, who critiqued the manuscript and served on my doctoral committee.

This experiment could not have been conducted without the cooperation and support from many people at The Park School. A special thanks to Dr. F. Parvin Sharpless, Headmaster, Allen Kershman, Principal, Anne Fogg, fourth grade teacher, and the parents and students of the 1983-1984 fourth grade class. I am also indebted to many people who assisted during the testing of my students, especially Linda Halpert,

Lorraine Schapiro, Jocelyn Burrell, and Barbara Hyman. A special thanks goes to Alice Morgan-Brown for her invaluable assistance coding the audiotapes used in the study.

A special thanks goes to Elizabeth L. Hatton for her excellent typing.

I am greatly appreciative of the support, love and encouragement which I got from my family and friends.

Most importantly, I thank David H. Bamberger, my husband, and Stephanie Ellen Bamberger and Jessica Margot Bamberger, my daughters, for their unrelenting patience, confidence, encouragement, and love.

DEDICATION

I wish to dedicate this paper to the memory of my beloved father whose love, support, and encouragement kept me going even when I grew weary and ready to give up. His pride in my every endeavor helped me to persevere.

TABLE OF CONTENTS

		Page
LIST OF TABLES		viii
Chapter		
I.	INTRODUCTION AND BACKGROUND OF THE PROBLEM	1
	Statement of the Problem	9
	Significance and Purpose of the Study	9
	Definition of Terms	13
II.	REVIEW OF RELATED LITERATURE	20
	Theoretical Bases	21
	Research on Problem Solving Processes	30
	The Potential of Logo	44
	Research on Logo	52
	Summary and Conclusions	62
III.	METHODOLOGY	65
	Purpose	65
	Sample and Sample Selection	65
	Instructional Unit and Materials	69
	Instructional Procedures	71
	Test Instruments	75
	The Posttest	77
	Hypotheses to be Tested	79

Design and Experimental Controls. . . . 80
Maturation. 81
Instrumentation 82
Mortality 82
Interaction 83
Dependent Variable. 84
Experience of the Testers 85
Coder Agreement 85
Statistical Analysis. 90
IV. RESULTS 92
V. INTERPRETATION, IMPLICATIONS AND
RECOMMENDATIONS 100
Problem Solving Strategies. 102
Problem Solving in the Classroom. . . . 105
Process Coding Sequence 106
Attitude. 107
Summary 110
Implications. 111
Recommendations 113
BIBLIOGRAPHY 116

APPENDICES 139

A. Example of How Children Use Logo . 139

B. Riedesel Inventory of Children's Attitude Toward Mathematics (RIATCM) 143

C. Traditional Mathematics Curriculum 151

D. Letter to Parents. 155

E. Logo Curriculum for Fourth Graders. 157

F. Problem Solving Posttest 185

G. Letter to Testers. 195

H. Interview Procedure for Posttest . 200

I. Revised Process Code Dictionary. . 205

J. Codings from Problem Solving Posttest 209

K. Individual Subcategory Scores from Riedesel Attitude Survey. . 225

LIST OF TABLES

Table Page

1. Breakdown of Stanine Scores on the Educational Records Bureau Test (1982). . . . 69
2. Instructional Time Periods for Experimental and Control Groups 81
3. Percentage of Coder Agreement Prior to Coding Posttest Date. 87
4. Percentage of Coder Agreement Following Coding of Ninth Student's Coding Sequence . . 88
5. Percentage of Coder Agreement Following Coding of Seventeenth Student's Coding Sequence. 89
6. Percentage of Coder Agreement Following Coding of Twenty-Sixth Student's Coding Sequence. 90
7. Number of Students Using Each Code on Each Problem. 94
8. Chi-Square Critical Values of Problem Solving Posttest. 95
9. t-Test on Mean Scores of Strategies Used Across All Problems. 96

10. Means and Standard Deviations and t-Test Scores of Each Subsection of Riedesel Attitude Survey. 98

CHAPTER I

Introduction and Background of the Problem

Thinking and problem solving are a significant part of a child's educational experience. Those in education have long been aware of the importance of solving problems. As noted by Polya:

> A great discovery solves a great problem but there is a grain of discovery in the solution of any problem. Your problem may be modest, but if it challenges your curiosity and brings into play your inventive faculties, and if you solve it by your own means, you may experience the tension and enjoy the triumph of discovery. Such experiences at a susceptible age may create a taste for mental work and leave their imprint on mind and character for a lifetime. (1957, p. v)

Recent research suggests that mathematics is essentially problem solving (Branca, 1980; Lester, 1977; Skypek, 1977; Sudyam, 1982). As a goal,

developing facility in solving problems is a primary objective for the study of mathematics. Mathematics education literature does not provide teachers with one accepted method for enhancing children's problem solving abilities, but many references suggest methods for improving children's problem solving strategies (e.g. Krulik & Rudnick, 1980; LeBlanc, 1978; Masalski, 1978; Musser & Shaughnessy, 1980; Polya, 1957).

These methods include teaching strategies which encourage the student to initially determine what is unknown and what information is given in a problem as a means of understanding the problem. Pictures or charts may also be drawn by the child to illustrate the problem, thus encouraging further understanding. Once the child understands the problem this approach to problem solving directs the child to decide upon a plan for solving the problem. Techniques which the child may employ include: recalling similar previous problems, familiar algorithms, or restating the problem in a way which clarifies it for the child. The child might look for patterns within the problem, use trial and error or work backwards in an effort to arrive at a solution. Once the plan is decided upon,

this approach to problem solving suggests that the child carry out the plan and arrive at a solution. When this is completed, this method directs the child to examine and recheck the solution.

Another method involves the teaching of programming to children using microcomputers. "Computer programming is an excellent means for developing problem-solving strategies" (Watt, 1982, p. 130). Similar to solving real-world problems, computer problems also have many different solutions and definite consequences. Programming enables children to learn how to handle information and provides them with examples of how society can be aided with the proper use of a computer (Charp, Bozeman, Altschuler, D'Orazio & Spuck, 1982). Programming with microcomputers promotes problem solving and logical thinking (Hatfield & Kieren, 1972; Johnson, 1983, Schoen & Blume, 1984). If used effectively, programming with microcomputers enables students to become better problem solvers (Kantowski, 1983). With a microcomputer, children are able to interact with, to test an hypothesis and view the results, and to direct their learning process by designing their own procedures. Many of the

strategies which are involved in problem solving (e.g. posing questions, analyzing situations, using trial and error, translating and illustrating results), are all needed when children program.

Logo is a high-level computer language which encourages children to think and solve problems by developing their own programs (Harvey, 1980; Solomon, 1984). Using Logo, children define a problem to solve and then build on their previous experiences in order to solve it. They must group their commands by sequencing their steps and may use subprocedures as steps to larger, more complicated procedures. They are also encouraged to think of their errors as steps useful in understanding their problem. The computer gently reminds them that human error is a very powerful impetus for learning (Olds, 1983).

When generating a project in Logo, children utilize many accepted mathematics problem solving strategies. They may begin by illustrating their project (drawing a picture) and then breaking the whole picture into workable parts. Muller (1983) writes, "In drawing pictures of houses, animals or just geometric shapes, youngsters gained a valuable experience in logical thought and problem solving"

(p. 55). Some children make charts to sequence their commands, while others just list the necessary commands. Trial and error (guess and check) strategies are frequently used, as well as those which use previously learned information. Finally, errors along the way allow children to deduce solutions to problems by analyzing given and missing information, enabling them to modify their procedures.

Dyrli writes:

> . . . perhaps most important are the contributions Logo can make to developing the thinking skills of children. According to the theories of Jean Piaget, the abilities to combine ideas systematically, hypothesize, predict consequences and consider the implications of actions in a new situation are all characteristic of the Stage of Formal Operations, which ordinarily begins around age 11 or 12. Such skills are used routinely in turtle graphics to define procedures, and create more complex procedures. Conversely, Logo also develops the important problem-solving strategy of breaking large problems down into manageable

> subprocedures. But whether working with procedures from the bottom up or from the top down, Logo promotes a spirit of experimental inquiry and trial-and-error learning that is truly unequalled. (1983, p. 62)

While making decisions on the student's own regarding the program created, numerous strategies are being used. "It is plausible, then, that a child's ability to think mathematically might be enhanced by learning to write programs, especially in an interactive computing system which provides immediate feedback" (Ross & Howe, 1981, p. 143).

Logo's origins are rooted in both computer science and Piaget's research on how children develop thinking skills (Papert, 1980). Papert and his colleagues believe that children learn by doing things and that a computer allows the child to manipulate and carry out exciting projects, giving the child a sense of mastery and power, confidence and achievement (Papert, 1982). This "user friendly" language motivates children to experiment and feel comfortable when using a computer. Piaget (1951) and others (e. g. Bruner, 1966; Malone, 1981) emphasize

the importance of intrinsically motivating children with play-like activities for many kinds of deep learning.

Research on Logo conducted at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology (MIT) has been anecdotal in nature (Berger, 1983; Davidson, 1983; Papert, Watt, diSessa & Weir, 1979; Watt, 1979). Based on records kept while children of varying ages and abilities worked on Logo projects, MIT researchers have concluded that using Logo improves problem solving performance.

Research on Logo is also in progress at the Bank Street College of Education in New York. Some of this research (Hawkins, 1983; Kurland & Pea, 1983; Pea, 1983; Pea & Kurland, 1983; Pea & Kurland, 1984) is less supportive of Logo as an aid to problem solving. These studies suggest little or no effect on children's problem solving strategies.

There has been limited empirical research on the effect of using Logo on children's mathematical problem solving strategies. Its use may have substantial impact on a child's attitudes about and ability to solve mathematical problems.

The computer, as noted by Ross and Campbell (1983), is already playing a prominent role in our society and "promises to profoundly influence practically all aspects of our lives in the future" (p. 109). Steffin (1983) suggests that the microcomputer's capacity for providing constant interaction between learner and machine, infinite patience even when answers are incorrect, and giving immediate feedback offers real promise for significant improvement in the classroom. In a recent edition of Computers in Curriculum and Instruction, the Association for Supervision and Curriculum Development emphasized the importance of "restructuring the curricular and methodology, if we are to provide students with experiences that will help them cope with the roles of today, as well as develop enough flexibility to obtain the new skills required for tomorrow" (Grady & Gawronski, 1981, p. vii).

Statement of the Problem

The purpose of this study was to determine the effect of Logo on the problem solving strategies used by fourth grade children. In particular, this study investigated whether children who were utilizing problem solving strategies as they programmed with the language Logo would use the same types of strategies when solving mathematical word problems without a microcomputer. The strategies of interest were a) guessing and checking, b) breaking a large problem into manageable parts, c) using diagrams or illustrations, d) using analytic or deductive reasoning to solve a problem and e) checking or looking back over completed work.

A secondary purpose of the study was to determine whether children who were programming with Logo would have a more positive attitude toward mathematics than those having no Logo experience.

Significance and Purpose of the Study

Teachers and administrators are eager to provide their students with experiences which give their students opportunities to solve mathematical problems and provide them with positive computer experiences. Many educators believe that the implications of computers for learning are great and that they are

"the most revolutionary education tool since the advent of chalk" (Rheingold, 1983). However, the Washington Post recently reported (1983) that a study by the Carnegie Foundation for the Advancement of Teaching found that educational material for computer instruction is greatly lacking. School systems seem to be purchasing equipment before planning an appropriate curriculum or hiring trained teachers. This lack of preparation results in the use of computers as programmed instructional materials, not as problem solving tools. Indeed, many who use microcomputers complain that the typical "Computer Assisted Instruction programs (CAI) leave students feeling that they are being controlled by the computer" (Judd, 1983, p. 20).

Taylor (1980) provides a framework for categorizing the various uses of a computer as a tool, tutor or tutee. The tutee approach has the child using the computer language to design games, drawings and programs on his own. The child must decide what to teach the computer; what to do next. The student is a participant in the work being done.

Logo has been described as the "first programming language that children could really use" (Rheingold, 1983, p. 38). Papert (1981) notes that

in the Logo environment the child is in control, not the computer. Programming in Logo places the learner in charge of:

- setting a problem to solve
- making choices
- playing with the problem, experimenting and trying out solutions
- building on what he already had done to do something more. (Watt, 1982, p. 112)

In On Knowing, Jerome Bruner (1962) emphasizes the importance of allowing the learner to manipulate and discover in a setting where the consequences of error are not overpowering. John Dewey (1900) also stressed the importance of allowing children to play in order to learn. Logo was designed to enable young children to control a computer allowing them to play, explore and learn (Abelson, 1982). It encourages children to explore and discover by using their errors to refine and rethink procedures. It fosters children's logical thinking and problem solving skills with the assistance of a nonthreatening and appealing "turtle". At the same time, Logo teaches through the implied mathematics of graphics rather

than overt mathematics (Papert, 1981; Watt, 1982; Wills, 1981).

Problem solving needs to be a part of the elementary mathematics program. Scores for the application of computational skills in problem solving are lower than those of computation based on data from the First National Assessment of Educational Progress (Carpenter, 1980). Carpenter, Coburn, Reys & Wilson (1976) conclude that "a commitment to working and thinking about word problems is needed for teachers and their students" (p. 392). In addition, children need to learn specific strategies and general procedures for solving problems (Sudyam, 1980). Specific heuristics to solve problems can be taught throughout the day, in mathematics as well as in computer classes.

It is essential to note that both in problem solving and learning to program a computer, one must describe a process or an algorithm and then test it (Ross & Howe, 1981). It is this parallel that teachers must capitalize on in order to help children acquire the skills and strategies needed to think through problems. An examination of the transfer of problem solving strategies by children who use Logo

may enable educators to modify their own computer and mathematics curricula to more successfully foster independent thinking and problem solving. This study developed a Logo curriculum which emphasized problem solving strategies appropriate for a mathematics curricula and encouraged logical thinking by students.

It is hypothesized that children's attitudes towards mathematics might be positively affected by their experience with Logo since they work with Logo with great willingness and ease. Thus, a secondary purpose of this study was to investigate whether children who were programming with Logo would regard mathematics as more exciting and more favorable after eleven weeks of instruction than those students who had no Logo experience.

Definition of Terms

This section will conceptually define the study's major constructs. Definitions appear in alphabetical order, rather than in order of importance.

1. Artificial Intelligence - a branch of computer science from which Logo grew. It gets its name from its attempts to simulate, using machines,

behavior that is usually regarded as intelligent in humans or animals.

2. Computer Aided Instruction (CAI) - "any educational endeavor that is aided by computers" (Goldenberg, 1982, p. 212); may use programmed tests, automated worksheets, simulations, or tutorials.

3. Debugging - a procedure which improves the behavior of a program that does not do what you want it to (Goldenberg, 1982) by taking apart and analyzing the procedures that make up a program.

4. Heuristics - a basic strategy, independent of any particular subject matter, which helps students approach and understand a problem so they can solve it.

5. Interactive Language - a language that involves giving the computer a command, by typing it onto the keyboard, and then having that command be carried out immediately. A response (feedback) is then supplied by the computer to the user. Changes are easy to make which facilitates programming.

6. Language - a way of communicating information or data between people or between people and machines. It is made up of a carefully defined set of rules and characters, so that meaning is achieved.

7. Logo - a high level computer language developed by Seymour Papert at the Artificial Intelligence Laboratory at the Massachusetts Institute of Technology. It is procedural, interactive and recursive. It contains both Turtle Graphics and list processing capabilities. Logo requires the learner to actively build on previous experience in order to solve problems. Logo is considered by some (Braun, 1983) to be more than just a language, but "an intellectual environment within which the student can discover mathematical and scientific principles" (p. 226). Harold Abelson writes:

> Logo is the name for a philosophy of education and for a continually evolving family of computer languages that aid its realization. Its learning environments articulate the principle that giving people personal control

over powerful computational resources can enable them to establish intimate contact with profound ideas from science, from mathematics, and from the art of intellectual model building. Computer languages are designed to transform computers into flexible tools to aid in learning, in playing, and exploring. (1982, p. IX)

8. Microcomputer - "a small computer system whose central processing unit (the part of the machine that can automatically carry out the instructions in a program) consists of a single chip or a small number of chips" (Charp, et al., 1982, p. 58).
9. The Park School - a non-sectarian, coeducational, independent, liberal arts school founded in 1912. The school has approximately 600 students from kindergarten through twelve and the campus is located on Old Court Road, in Brooklandville, Maryland. (A detailed description of the school, its philosophy and population is given in Chapter III, Target Population).

10. Problem Solving - a process of understanding a problem and then devising, testing and examining a solution for a given problem.
11. Procedures - groups of commands (making something happen) and operations (resulting in a product or information) that are combined together and named by the user. Procedures can be used as steps in other procedures. Each procedure has a title, a body which defines it, and an end statement.
12. Process Problem - a problem which emphasizes the procedures inherent in problem solving using a variety of strategies (LeBlanc, 1977).
13. Programmer - "a person who analyzes a particular job to be done on a computer and specifies a sequence of instructions to do the job" (Crowley, 1967).
14. Recursive - a language is recursive if a procedure can be a subprocedure of itself, allowing very large problems to be stated in very compact ways.
15. Riedesel Inventory of Children's Attitude Toward Mathematics (RICATM) - a general attitude survey developed by Alan C. Riedesel in 1961. It uses

a four-choice Likert Scale to collect the feelings that children have toward mathematics (See Appendix B).

16. Strategies - plans developed to solve a problem.

17. Traditional mathematics curriculum - a curriculum which has, as its foundation, the principle that the process of mathematics is more important than the final product that is obtained (See Appendix C for a detailed analysis of this curriculum for fourth graders at Park School).

18. Turtle - a computer-controlled "cybernetic animal" that "lives" on the display screen and responds to Logo commands that make it move and rotate (Papert, 1980). It may, at the user's discretion, leave a trail or line behind it during the creation of drawings.

19. Turtle Graphics - a subset of the Logo language which permits the user to draw geometrical designs by directing the computer to change the "heading" and "position" of a small triangular screen object that is referred to as a Turtle.

20. User Friendly - a computer system or language which is designed to be as easy for a novice user to implement (sufficient instructions to work through) as it is for an experienced user.

CHAPTER II

Review of Related Literature

The 1980's have witnessed an emergence of interest in two areas of mathematics education -- problem solving and the use of computers. While each has received its own publicity, recent research has been studying the relationship between these two areas. The focus of this study was to explore the impact of a computer language called Logo on the problem solving strategies used by young children.

The rationale and significance of a study are dependent upon its theoretical bases. The review of the literature for this study is organized as follows. First, the theoretical bases of the development of Logo will be addressed. Then, the research surrounding problem solving and its role in the mathematics curriculum will be discussed. Following this, literature related to the development and initial evaluation of Logo will be reported. The concluding section will present a review of studies that investigated the effect of Logo on problem solving.

Theoretical Bases

The bases for this study are the learning theories of Jean Piaget and Jerome Bruner. "Meaning theory," as espoused by William Brownell, offers a rationale for this study's emphasis on children enjoying and understanding the mathematics they are using. Finally, George Polya's theories on problem solving provide the framework for the instructional approach used in this study.

Jean Piaget created a model for understanding the development of children's cognitive processes. This model is developmental in nature, yet the age for entering a given stage may vary between individuals. The order of these stages, however, remains constant. Piaget's stages include the sensorimotor, preoperational and operational stage (which is separated into concrete and formal operations). These stages concerned themselves with the general psychological processes which foster cognitive growth. Piaget believed that a child's advancement through the stages was affected by the child's experiences, social interactions, overall maturation and equilibrium.

Inspired by this model of cognitive development, Seymour Papert, and a group of researchers, developed a

computer language which combined Piaget's principles with the capabilities of artificial intelligence in order to allow the learner to build his own intellectual structures through his experience (Papert, 1980). Piaget's ideas that "children are builders of their own intellectual structures" (1952, p. 7) prompted Papert to develop the Logo language in such a manner that exploration and discovery were major aspects of its use.

Piaget emphasized the importance of numerous, varied experiences for children. He felt that children developed the concept of number, as well as that of time and space, through the manipulation of objects and through multiple experiences using all of their senses. Piaget (1952) hypothesized that the activities carried out by the child were critical in promoting the growth of intelligence from infancy to adolescence. He argued that children use physical actions on objects to acquire an understanding of these objects. Children, stated Piaget, acquire and assimilate new information through exploration and experience. Papert's translation of this was that children could manipulate and explore the Logo language and in doing so could program the microcomputer. Papert believes that with a minimal amount of information, and

with time to explore, discover and manipulate, children will learn how to teach the computer how to think, that is to program the computer to accomplish a task (Papert, 1980). Through this process of exploration and manipulation, Papert believes children will come to understand how they themselves think. Thus, in Papert's opinion, teachers do not need to develop a structured curriculum for teaching Logo; rather, teachers should foster the children's natural tendency to experiment within the Logo language.

Another premise behind Piaget's theories is that of respect for individual differences. According to Piaget all children pass through the same stages but at different rates. These differences may be due to influences of maturation, genetic inheritance, and environment. Papert recognizes this in his philosophy of Logo, expressing the need for teachers to teach only those commands and concepts that the child is ready to understand. In initial studies of Logo, with students from the Brookline school system, the children were only taught the commands which would help them with their individual projects. Teachers observed the child's style of learning and methods of programming, allowing students to proceed at

their own pace, in their own way, in their own style, and with their own approach (Papert, Watt, DiSessa & Weir, 1979).

The significance of childhood experiences was further emphasized by the importance which Piaget placed on stimulating children through play-like activities (1951). Piaget stated that play-like situations would motivate children to spend more time and effort learning. Papert's creation of an appealing manipulative "turtle," which was later modified to a "turtle" which could move around a computer monitor screen, is an application of Piaget's concept of play and learning. Using this "turtle," children explore and learn in an independent manner which, Piaget argues, fosters intrinsic motivation. Papert wrote, " . . . the strength of the computer, is its ability to help children think--to help them understand complicated concepts by placing them in a context that is relevant to children" (1982, p. 86).

Piaget noted that children need to interact socially, learning how to evaluate their thoughts and ideas by comparing them to others (Wadsworth, 1971). Papert incorporated this principle when he discussed the ideal way to teach children Turtle Graphics, namely, for children to work together when planning projects, to talk

and solve problems aloud and to share ideas and findings whenever possible. Thus, Papert incorporates Piaget's belief in the importance of interaction as the key to fostering learning with Logo.

Piaget's writings emphasize the importance of language in facilitating learning. Furth & Wach (1974) write that children need to be educated so they will one day be capable of using language in an intelligent manner, thus further developing their intelligence. Piaget's emphasis on language is exemplified in the language-like Logo commands and in the reliance on verbal interaction in the teaching model promoted by Papert. Logo commands are simple terms taken from words in a child's vocabulary. When sequenced, these commands enable the child to communicate with the computer in a highly sophisticated way; however, it is the child's inherent ability to use language which enables the use of Logo.

A basic premise in the philosophy of Logo is that children should use the materials around them as inspiration to create interesting images on the screen. They are encouraged to create drawings of objects that they have seen or that interest them. This is an introduction to computer technology in a manner that has

significance to children at their particular stage in life. It parallels John Dewey's (1900) emphasis on the importance of making education like life itself. As the child interacts with the computer, his ideas are fostered and as he learns to program, his learning becomes active and self-directed. The child desires to produce increasingly difficult screen images and thus becomes intrinsically motivated to try newer and more difficult projects.

Jerome Bruner's (1966) theory of intellectual growth postulates that learning, which passes through three stages (iconic, enactive and symbolic), is primarily reliant on the actions children use to understand the world. Bruner emphasizes the need for children to feel that the activity they are doing is worthwhile and meaningful. He implies that the task for education is to provide experiences, and means of translating these experiences into ideas which the child will find meaningful. Manipulation and action are seen as necessary conditions for learning. Consequently, intellectual growth and mental growth will be dependent upon the environment in which one is raised and the experiences to which one is exposed. According to Bruner's theory,

computer experiences which permit children to control the machine not only motivate, but also foster intellectual growth.

> Growth is characterized by increasing independence of response from the immediate nature of the stimulus. Growth depends upon internalizing events into a 'storage system' that corresponds to the environment. Intellectual growth involves an increasing capacity to say to oneself and others by means of words or symbols, what one has done and what one will do. (1966, p. 5-6)

Papert's theories on children's use of computers corresponds to Bruner's suggestion that the study of mathematics should be made analogous to a puzzle form, that is a "game in which there are a set of givens and a set of procedural constraints, all precisely stated" (Bruner, 1962, p. 97). Bruner (1962) stated that when children discover they acquire a sense of confidence which is, in itself, rewarding. With Logo, educators lead students to develop an understanding of commands which will allow them to perform operations that have meaning

for them. In addition, the feedback which the Logo language provides enables children to understand what has occurred.

Bruner also postulates that children acquire new learnings when they are not afraid of the consequences of making an error (1962). With Turtle Graphics, errors are seen as natural, necessary and a non-threatening aspect of using the language. Children learn that their mistakes may promote a new or different creation from that which they had originally intended or that their mistakes may identify the error in their program. There is an aura of playfulness because the consequences of error are not overpowering. This aura may motivate children to experiment in ways which might otherwise have been too frightening.

Bruner's (1966) theory also emphasizes the need for children to have knowledge of the results of their work in order to know whether they were reaching the goals that they had set for themselves. This type of feedback has become an essential part of the Logo experience. "Because Logo is a vehicle for free exploration, knowledge built from Logo is syntonic, appropriate to the person and experienced as an authentic, intimate part of the self" (Lawler, 1982, p. 150).

William Brownell (1935) advocated the need for children to see "sense" in what they learn, and hypothesized that if children understood, learning would improve. He believed that children need to understand and identify the relationships necessary to the solution of a problem (1942). This meaning theory supported the emphasis on understanding how and why mathematical operations worked, rather than merely memorizing formulas and principles. Papert's emphasis on allowing children to use the experiences in their lives to create meaningful graphic designs meshes with Brownell's theory. Here again the emphasis is what is meaningful and important to the child. Commands make sense because they are part of the child's natural language; procedural names are meaningful because they are created and defined by the child.

Using Logo arouses children's curiosity, motivating them to achieve competence in things that they do. Logo learning is active learning, shifting the emphasis from a teacher-directed to a student-centered classroom, utilizing the child's natural curiosity (Masalski, 1978). In addition, the child sees meaning in what she is doing, and this meaning facilitates learning.

Brownell further emphasized the need for solutions to be clearly summarized for the child. Once a problem has been solved, the child needs to be able to understand and identify the components useful in solving the problem. When Logo problems are being solved, children often break the problem into manageable parts. Once a solution is reached for each part, they can clearly identify and merge the solution's components as a whole.

In summary, exploration, discovery, manipulation and feedback, as well as their interrelationship and the interaction children have with the environment, in a personally meaningful, non-judgmental manner are all emphasized by the theorists discussed above. Logo's creators utilized these principles as theoretical support for their computer language.

Research on Problem Solving Processes

During the past several decades, education journals have witnessed an increase in articles and research dealing with problem solving in mathematics. While this is not a new area of mathematics education, increased attention has been focussed on it.

Mathematics education has its trends. In the 1960's the trend was "New Math," and young children began learning set theory in order to make sense out of the arithmetic being taught to them. In the 1970's there was a dual concern about the need to return back to basics or to continue stressing the meaning behind mathematical operations. In a special session during the 1976 annual meeting of the National Council of Supervisors of Mathematics (NCSM), a listing of ten basic skill areas was developed to provide direction for mathematics programs. Problem solving is the first item on that list:

> Learning to solve problems is the principal reason for studying mathematics. Problem solving is the process of applying previously acquired knowledge to new and unfamiliar situations. Solving word problems in texts is but one form of problem solving, but students also should be faced with non-textbook problems. Problem solving strategies involve posing questions, analyzing situations, translating results, illustrating results, drawing diagrams, and using trial and error. (1977, p. 20)

In its Agenda for Action (1980) the National Council of Teachers of Mathematics recommended that "problem solving be the focus of school mathematics in the 1980's" (p. 1). This organization suggested that mathematics programs should teach the definition and language of problem solving, as well as strategies and applications for reinforcing problem solving activities. Following this recommendation, the Priorities in School Mathematics (PRISM) project (1981) surveyed groups of people involved in education to collect information and reactions to mathematics curriculum changes for the 1980's. The PRISM data indicated that problem solving was consistently ranked as an area to be emphasized in mathematics education.

The 1980's were officially declared the decade of problem solving.

Problem solving as part of the K-12 mathematics curriculum did not originate in the 1980's. While the first edition of George Polya's book, How to Solve It (1957) has influenced researchers and educators since the late 1950's, this was not the first document to discuss problem solving. Pappus, a Greek mathematician who lived around 300 A. D. wrote about heuristics to solve

problems. Rene Descartes, who lived from 1596-1650, began writing a method for solving problems in his unfinished, Rules for the Direction of the Mind; and Bernard Bolzano, who lived from 1781 - 1848 wrote extensively about logical thought (Polya, 1957). William Brownell (1942) emphasized the need to have problem situations clearly stated and well within the child's understanding. He specified the need to provide children with different types of problems which would encourage the child to search beyond the solution of the problem to an understanding of what had been done and why. The importance of problem solving in today's mathematics curriculum is being felt by educators and researchers alike.

Polya's philosophy about mathematical problem solving encouraged teachers to challenge their students by creating problems which are appropriate to the age level of the child. By doing this Polya held that the child would be given an opportunity to engage in independent thinking. If teacher assistance was unobtrusive and discrete, the student's confidence in his or her ability to perform independent work would grow. Consequently, Polya viewed the teacher's role as that of helping

children solve a given problem while providing them with tools to solve future problems (Polya, 1957).

This philosophy parallels Seymour Papert's philosophy regarding Logo. Children are challenged to create projects of their own design, to think logically about sequencing their steps and to work independently, with a minimal amount of teacher intervention. It is Papert's contention that in this environment computer skills will develop and children will be prepared to use the technology of the future to solve problems.

Despite the numerous articles which have been written, the differing definitions and approaches which have been offered, and the research studies which have been conducted, the area of problem solving continues to generate more questions than answers for investigators. In particular, the relationship between problem solving and the ability to apply specific processes of problem solving to the solution of problems is not clear. "Past studies have been thought-provoking but have not provided conclusive evidence about the relationship between conceptual knowledge and processes and how they relate to problem solving abilities" (Webb, 1979, p. 84). A study by Kantowski (1975) suggested that the ability to use

heuristics, as well as to recall important important content, may be related to problem solving ability. In a study by Simon (1975) it was implied that heuristic problem solving capabilities must be a part of an effective problem solver's repertoire of skills. Therefore, the focus of this study was the processes used by students when solving problems rather than the products obtained by those processes.

As a process, problem solving is defined as applying previously learned information to unfamiliar situations (Branca, 1980). In this interpretation one looks at the methods or strategies used to solve the problem. Problem solving becomes a process involving an understanding of why a particular approach to solving the problem worked (Krulik, 1977). Process problems emphasize the procedures inherent in problem solving and can be solved using a variety of strategies (LeBlanc, 1977). The importance of understanding the processes necessary to solve a problem were emphasized by Brownell (1942), "A problem is not truly solved unless the learner understands what he has done and knows why his actions were appropriate" (p. 14). It is the process of solving the problem that allows children to understand what the problem means so that they

can develop a solution. "Process problems are used to encourage the development and practice of problem solving strategies. They provide an opportunity for students to devise creative methods of solution, to share methods with other students, and build confidence in solving problems" (LeBlanc, Proudfit & Putt, 1982, p. 105).

The building of confidence is extremely significant in that it encourages the child to take risks and enjoy mathematics. Consequently, one of the aspects of teaching problem solving that educators are encouraged by is the tendency of students to enjoy the process of challenging puzzles and solving different types of problems. Children are able to apply their learning to functional activities which help them to see the applicability of their math skills. Given real problems which are related to their immediate experiences, problem solving is a more effective means for encouraging students to think about issues (Glenn & Ellis, 1982).

Research on problem solving suggests that children are more successful solving problems that interest them, contain familiar settings, have the question at the beginning, rather than the end of the problem, and contain one, rather than multiple steps (Sudyam & Weaver, 1977).

Problem solving performance is enhanced by teaching children to use a variety of heuristics or strategies when solving problems (Sudyam, 1980). Finally, since problem solving requires a great deal of decision making, educators are realizing that students need opportunities to make decisions in the classroom (Thorton & Bley, 1982).

Polya's influence in the area of problem solving is evident by the numerous authors who discuss, in slightly different terms, the same four processes that he reported would aid children when solving problems (e.g. Krulik, 1977; LeBlanc, 1977; Masalski, 1978; Schoenfeld, 1980; Wickelgren, 1974; Worth, 1982). He suggests that educators attend to the heuristics as well as content when teaching problems. Students, he feels, need to understand the procedures of a solution, not just arrive at it.

The strategies which Polya and others stress require that the student first read and understand the problem. Once this has been done the solver must devise a plan. This can be done with drawings, if necessary, to separate what is given and what is unknown. A previous experience with a similar problem might be used to facilitate the development of a plan. Once the plan is developed, the student is to carry out this plan, checking each step

along the way. Like the trial and error approaches used to investigate Logo programs, the solver must not only try different approaches, but repeatedly do this; perhaps adapting the trials to the circumstances. If the solution fails, another plan should be in reserve. Once the solution is found the student needs to reflect back on what has been done, to examine the problem and decide if it is a reasonable solution.

Often, random trial and error (equivalent to random sampling with replacement from the population) is employed when children first attempt to solve a problem. But "to avoid going around in circles it is obviously desirable to remember what sequences of actions have already been tried with no success" (Wickelgren, 1974, p. 46). By writing things down, children can be sure that all sequences in a problem are generated.

In a study conducted by Kantowski (1977) eight ninth-grade algebra students were acquainted with the use of heuristics in problem solving and then given instruction in geometry using heuristic instructional techniques. A posttest consisting of geometry and verbal problems was administered to each individually. Subjects were asked to "think aloud" as they solved problems and

their verbalizations were later analyzed according to a coding system developed by Kilpatrick (1967). The results of this study indicated that regular patterns of analysis and synthesis followed the use of heuristics and the tendency to use heuristics increased as problem solving ability developed.

In a study of 11 college students enrolled in a problem solving course (Schoenfeld, 1982), it was concluded that their performance, as measured by test scores on non-routine problems, was greatly enhanced by their general instruction in heuristics. When compared with a control group of 8 students, the experimental group used many of the problem solving heuristics and managerial strategies that had been taught during their month long course. Schoenfeld inferred that the students transferred their heuristic behavior on problems which were in some way related to problems they had studied previously.

One strategy which Polya (1957) stresses involves arranging the steps of the problem in a sequential order and then working backwards in order to visualize the final solution. If taught to children during Logo instruction, these same steps enable children to use the programming language with relative ease.

In a computer and problem solving study (Schoen & Blume, 1984) eighth grade students were divided into a BASIC programming group and a non-programming group in an effort to investigate the effects of computer programming on students' ability to solve mathematical problems and on the processes they used to solve such problems. Following programming instruction, students took a two-part paper and pencil test. The first part included 15 multiple choice items from the Iowa Problem Solving Test and the second part included 18 researcher-developed test questions. The second part was meant to measure the ability to apply the sequential logic that computer programming entails. A "think-aloud" approach was used to solve five problems and an adaptation of a process coding system developed by Kulm (1981) was used to code each audio-cassette. The findings suggest that BASIC did not affect the student's scores on the paper and pencil problem solving test. Schoen & Blume did state, " . . . if this research is to identify an effect of computer programming on problem solving performance, it will be in terms of the processes the students employ" (p. 15).

Webb (1979) also used a think-aloud procedure to analyze the heuristics that 40 Algebra II students used

when solving eight problems. The reasoning for this type of procedure was,

> The thinking aloud procedure provides an edited version of the thoughts the solver uses while solving a problem. The number of processes and their frequencies identified by this means provides an approximate measure of the processes used while solving the problem. This is the best method known to observe such processes. (1979, p. 84)

Findings from this study indicated that an interrelatedness existed between conceptual knowledge and the processes that students use. It was also the researcher's feeling that students needed to acquire heuristics in order to solve problems.

Another team of researchers used a process-sequence coding system in an effort to understand the heuristics used by children when solving mathematical problems (Lucas, Branca, Goldberg, Kantowski, Kellogg & Smith, 1980). Audiotapes were used, along with the subject's written word and the interviewer's notes to order the strategies used by the child. Similarly, a process-coding

sequence was used by Kulm et al (1981) to evaluate the effect of age and sex on the types of problem solving strategies used by students.

The disadvantages of the "think aloud" approach are that some children cannot verbalize what they are thinking about, some are distracted when asked to talk out loud, and some may chatter without ever making their ideas clear. Nevertheless, this technique is widely used today to analyze the thinking processes used by children as they solve problems.

In a review of the results of the National Assessment of Educational Progress, Carpenter, (1980) reported that mathematics problem solving exercises which required students to do more than add, subtract, multiply or divide caused considerable difficulty for 70,000 9-, 13- and 17-year-olds. Non-routine but simple problems (e.g. "Mr. Jones put a wire fence all the way around his rectangular garden. The garden is 10 feet long and 6 feet wide. How many feet of fencing did he use?")(Carpenter, 1980, p. 562) demonstrated that application of skills was a difficult task for these students.

Instruction in problem solving requires the identification of problems which are more than the basic

textbook problems or exercises which require only a one-step plan to solve them. For a problem to be effective, it should have three attributes. First, the solver has to have an involvement with a situation where a solution is needed. Secondly, the path toward solving the problem is unknown. (After all, this is what is meant by a problem in the first place). Finally, there is no one algorithm recognized by the solver that can be used to solve the problem (Masalski, 1978). Programming in Logo incorporates these three characteristics into the procedures necessary to create interesting graphic designs. Children must first decide upon a project, they must define the design that they wish to draw. Secondly, they need to develop the sequence of commands that will generate this drawing. Finally, they may use any one of a number of strategies, (e.g. trial and error, logical deduction or drawing a figure) to solve this problem. This further emphasizes what Brownell (1942) felt was an important element to consider about mathematics -- the learner needs to understand what has been done to see these actions as appropriate.

In summary, the writings and research by mathematics educators suggest that the teaching of problem solving is

an important element in the mathematics curriculum. By analyzing the processes necessary when solving problems, students will acquire a deeper understanding of mathematics. In addition, heuristics can be taught directly to students and these may be used by them to solve other problems.

The Potential of Logo

Educators have characterized the potential of computers as being exceptional. One (Uhlig, 1982) feels that there has been no phenomenon since Sputnik which will impact on education as much as the microcomputer. Although the rationale supporting this potential varies, one assessment offered by Papert (1980) noted the "holding power" which the computer has for children. Children of all ages seem eager, enthusiastic, and motivated to use this new piece of educational equipment. The microcomputer presents the child with challenging problems and allows him to create artistic works. It also encourages a spirit which characterizes faulty programs as unfinished projects rather than as mistakes, placing the learner in control (Papert, 1973). The computer may also stimulate children to apply problem solving techniques.

"It requires a tremendous amount of problem solving ability to make a computer solve a problem. It requires the kind of analytical thinking we want our students to learn" (Norris, 1981, p. 26). If the role of education is to foster thinking, to encourage children to make decisions and to solve problems independently, thus preparing them for the future, then the microcomputer may be a valuable tool.

> Educators will help their students by providing them with the kinds of skills they need to function successfully in a computer based society. Computers can be used to promote those aspects of learning for which they are best suited, freeing teachers to work with students in ways a computer cannot. The potential for improving the quality of life is good. (Sturdivant, 1981, p. xiv)

Logo, a computer language developed by Seymour Papert and his colleagues at the Massachusetts Institute of Technology, is designed to give children responsibility for defining a problem to solve, experimenting with solutions, building on previous learnings and taking

charge of their computer experience (Watt, 1982). Using Logo, children program the computer to develop and test their own models and learn to debug procedures which cause them difficulty. They must systematically sequence their commands and then teach them to the computer.

The Logo language is frequently introduced through Turtle Graphics, which permits children to draw by means of a triangular cursor ("turtle") which may be moved about a monitor screen, following the commands entered by the child. "Programming the turtle begins by making one reflect on how one does oneself what one would like the turtle to do" (Watt, 1982, p. 87). Papert (1980) suggests that by discovering how commands are combined to move the turtle, children learn about planning, problem solving and programming, as well as developing intellectual skills which help them think mathematically. Children must apply learned mathematical principles to solve problems and this emphasizes the "process of mathematization" (p. 194) by giving the child living mathematical experiences.

The characteristics of Logo have made it a very appealing language for school use. The language is interactive, so once a command is typed in, it is carried out immediately. It is procedural, allowing students to

define new commands which, when combined, create a program. It is recursive, allowing a procedure to become a subprocedure of itself. These features permit large problems to be viewed in compact ways. Logo is "user friendly," providing error messages which are sufficiently detailed to permit identification and correction of "bugs". At the same time, as noted by Harvey, (1982) "Logo has no threshold and no ceiling. This means that Logo is easy enough for anyone to use, but is powerful enough for any project" (p. 182). Morgan (1982), Editor in Chief of Byte, believes that "the first computer language you learn has a lifelong effect on how you think, computerwise. Thus, the computer language we choose in the schools becomes vital" (p. 6). Although Logo was initially developed to provide young children with opportunities to program a computer using their own personal experiences, Papert's contention was that by programming, computers would become tools which could positively stimulate the mind of the child. His emphasis was on programming the computer, rather than using pre-programmed software (Papert, 1981).

In his book, The Computer in the School: Tutor, Tool, Tutee, Taylor (1980) suggests that by placing children in

the position of tutor and the computer in the position of tutee, children learn to shift their focus of attention away from the end product towards the process of problem solving. They become more interested in the manipulation and understanding of facts rather than merely in the acquisition of facts. Children who program a computer are teaching it how to think and are, in turn, exploring how they themselves think. Thus, children become active participants in their learning, which gives them a sense of power and control and a feeling of mastery (Papert, 1980).

After its inception, Logo became more than just a computer language; it became an entire philosophy of education. "This philosophy holds that children learn best when they are free to discover relationships for themselves, rather than when they are told what to look for. It values active and self-directed exploration over direct and explicit instruction" (Hassett, 1984, p. 25). Papert envisioned schools which would allow children free access to explore, and Logo was seen as the vehicle to make this possible (Papert, 1971).

In assessing the benefits of computers in general, perhaps the greatest benefit that they have in mathematics

education is as a programming or problem solving device. Gawronski (1981) compares the writing of computer programs with the practicing of problem solving strategies. By constructing computer programs children must think about their problem and the steps necessary towards a solution of it. "Learning to program, in itself, is an exercise in problem solving, for the student goes through all the steps of the problem solving method" (Schiffman, Tobin, & Buchanan, 1982, p. 567). Due to the microcomputer's interactive nature, students can then test the consequences of their actions. By analyzing their coded problem solving steps and interpreting any errors, students may then modify their procedures.

Thus, Logo may be an appropriate problem solving tool. It not only gives children mastery over a machine, but it facilitates the use of estimating skills, geometry skills, sequencing approaches, and problem solving strategies.

One problem solving strategy utilized by children while programming in Logo is planning. Turtle Graphics projects are planned, prior to using the computer, by drawing sketches on paper. Children then begin breaking down their projects into manageable parts which may be

readily translated into the programming code. At the same time, both successful and false starts are recorded because "to avoid going around in circles it is obviously desirable to remember what sequences of actions have been tried already with no success" (Wickelgren, 1974, p. 46). Hence, planning involves recognition and reflection upon prior approaches toward solving this or a similar problem. Wickelgren (1974) suggests writing down the steps which work so that difficult sequences of action can be generated.

Another valuable strategy that is emphasized while children work on programs in Logo is that of "debugging" faulty procedures. Children learn the necessity for communicating precisely with the computer in order for it to carry out their commands. Explicit and literal commands need to be developed or problems arise. Yet, these problems are often reflected upon and then corrected, since feedback is automatic and error messages are precise. Children begin developing the skill of using errors to solve problems. They continually describe, test and compare their outcomes.

As children learn to program successfully they begin developing the logical and intuitive aspects of their

minds. Previous procedures get used as subprocedures for new projects, and trial and error is utilized less frequently as experience permits children to solve more sophisticated problems.

At the Artificial Intelligence Laboratory and within the Brookline Schools, in Cambridge, Massachusetts, Papert and his colleagues studied the effect of Logo. In order to verify and experiment with the effectiveness of his new language Papert applied for a grant from the National Science Foundation (NSF). He wrote, in the introduction of his proposal:

> . . . technology is used not in the form of machines for processing children but as something the child himself will learn to manipulate, to extend, to apply to projects, thereby gaining a greater and more articulate mastery of the world, a sense of power of applied knowledge and a self-confidently realistic image of himself as an intellectual agent. (1971, p. 1-1)

Papert conducted funded studies in the Boston-Cambridge area involving children representing a variety of settings and ages. Research from these initial

projects and those which followed will be elaborated upon in the next section of this chapter.

Research on Logo

Initial Logo research focused on the effectiveness of the language as a problem solving tool for children. Findings over a 14-year period at MIT suggest that intuitive, logical thinking, previously believed to be developmentally acquired by children at age 12, could be observed in children much younger. More current research from the Bank Street College of Education's Center for Children and Technology refute these findings and challenge the credibility of these claims. Thus, results of existing Logo research are inconclusive.

Perhaps the study reflecting the largest scope and the most documentation was that of elementary school children in Brookline, Massachusetts, during the 1977-1978 school year. Although all of the sixth grade students in a selected school learned Logo, careful anecdotal records were kept on only 16 of the students. These children represented a heterogeneous group in terms of intelligence levels (as reported by teacher reports and test results on standardized achievement tests). These children worked in

small groups in an individualized learning environment with trained Logo teachers, as well as regular classroom teachers. No pressure was exerted by the teacher or by other children to have these children perform in specified ways. Teachers kept careful records noting the students' progress, general attitudes about their work, and their basic learning styles within the Logo environment. Individual profiles of the 16 children explained how a project-oriented environment was created for each child (Watt, 1979). By the conclusion of the study, even students with severe learning problems had made gains in their learning. The researchers concluded that "programming in Logo bears on the important problems of bringing out and enhancing individual cognitive styles" (Papert, Watt, diSessa & Weir, 1979, p. iii).

One issue which was examined in the Brookline project was whether the knowledge gained in the Logo environment would transfer to other instructional activities. While this was not probed in detail, the researchers suggested that skills and principles in Logo were integrated by students into other activities (e.g. improvement in their ability to estimate angles)(Papert, et al., 1979). They also suggested that the positive computer experiences

could transfer to a more positive attitude toward mathematics in general.

Case histories were recorded on students who participated in a seventh-grade teaching experiment at a junior high school in Lexington, Massachusetts. In contrast to the type of work observed in mathematics classes, children were eager and motivated to continue their Logo projects although completion required consistent effort over many days or even weeks. These researchers saw this self-motivated persistence as promoting the children's developing self-image of themselves as "active intellectual agents" (Papert, 1971, p. 8-1).

Similarly, in their study of a second grader who had learned the basic Logo commands in first grade, Cynthia Solomon and Seymour Papert (1976) found that even a young child could work with sustained interest and a high level of independence and determination when solving problems using the computer and Logo.

Perlman's (1974) Logo work with children between the ages of 3 1/2 - 6 1/2 suggests that even young children can learn to program the computer to enable it to perform simple commands. More amazing was her finding that

interest and motivation remained high, even after a two hour period.

Papert noted a similar example of high level involvement and sophistication with 15 highly motivated ten- to twelve-year-olds in Exeter, England (Papert, 1973).

In each of these studies the researchers concluded that children worked with a high level of independence and persistence when programming in Logo. In addition, children of varying ages and abilities made gains in their learning.

Schwartz (1984) used 22 five-year-olds in a Logo study to determine their decision making skills. During the eight weeks, 90 minute sessions were planned and audio/video taped. Findings suggested that cumulative learning was evidenced and interest and ease around the computer was high.

Robert Lawler's doctoral dissertation interpreted his eight-year-old son's experiences using Logo over a six-month period at MIT's Artificial Intelligence Laboratory. Focussing on his son's fascination with symmetry, Lawler posed problems while Logo designs were being created at the computer. These problems were designed to encourage the child to postulate and to

subsequently test his theories on symmetry via Logo. Lawler only taught new aspects of the Logo language when it made solving a particular problem easier. No fixed curriculum was followed; rather Lawler introduced necessary programming concepts at opportune times. Lawler concluded that his son understood and enjoyed the entire experience. However, six months later, the benefits of the Logo experience became manifested when his son solved a complicated mathematical problem from school, which was not computer based, using Logo strategies. As noted by Lawler, "Robby's explanation witnesses that he conceived of his exploration in terms of that past Logo experience" (1980, p. 18).

A study conducted with fifth-grade students in a public school in Syracuse, New York attempted to observe the effects of learning Logo on the logical reasoning ability of the students. Students were randomly assigned to a control or experimental group. The students in the experimental group learned Logo, while those in the control group received no special instruction. The results demonstrated that students in the experimental group interpreted recursive statements better than the control group (Seidman, 1981).

Milner (1972) taught Logo to 38 fifth-grade children to determine whether Logo programming affected a child's performance in mathematics. While it was found that these children acquired a "notion of a variable" through programming, the researcher also found that certain problem solving behaviors were developed. This occurred in students regardless of their differences in ability, as measured by their Standard Achievement Test scores.

In a summary of various studies of Logo with young children, Howe, O'Shea and Plane (1979) concluded that students who had used Logo were better able to argue sensibly about mathematical issues and explain problems more clearly while simultaneously learning about the nature of polygons.

In each of these four studies researchers noted benefits in children's ability to think logically and solve mathematical problems using an understanding of mathematical concepts.

At the Phoenix School in Cambridge, Massachusetts, Davidson (1983) observed transfer from the debugging of Logo programs to the solution of why a word processor had printed gibberish. He wrote,

> I would like to think that any school that fosters problem-solving, especially in an environment with computers, will bring out a natural ability that most of us have. It's just that most of us haven't been encouraged, haven't even been given the opportunity to use that ability. (p. 6)

An account of how three handicapped children from the Cotting School for Handicapped Children learned to debug faulty procedures, enables the reader to see how a very positive message is conveyed to these children. Berger (1983) writes, "In the Logo learning environment, these bugs are often catalysts for mental and emotional self evaluation. The process of debugging requires reflection on and correction of one's own thinking" (p. 1). These children came to see in a playful, nonthreatening way that making mistakes is a human characteristic different from experiences they had encountered before.

Both Berger and Davidson conclude that the process of reflecting upon one's errors (an important strategy employed when using Logo) enables one to locate problems as well as understand that errors occur in most people's work.

Clements and Gullo (1984) conducted a twelve-week study using 18 six-year-olds to assess the effects of computer programming on divergent thinking. Their findings demonstrate that the programming group had higher levels of reflectivity and divergent thinking. Although the study involved only a small number of students, its conclusions suggest that Logo programming may increase problem solving abilities. Similarly, Statz (1973) noted significant differences in pre-test and posttest scores of fluency and originality, as well as overall divergent thinking by ten-year old children who had 12-18 hours of Logo instruction. Hence, Logo programming may increase some aspects of problem solving ability and may affect a child's cognitive style.

Each of these studies demonstrates the positive effect that researchers have attributed to the learning of Logo. These hypothesized effects include a more positive self image, increased motivation and interest in mathematics, as well as a positive effect on one's problem solving abilities. However, recent work at Bank Street College of Education's Center for Children and Technology concludes that the beneficial claims for Logo have been exaggerated.

Pea and Kurland (1984) noted a multitude of factors which affect a child's ability to learn to program. Among these are the child's intellectual capabilities, the teacher's capabilities, the environment in which it's taught and the appropriateness of the activity to the child. They question whether there is significant evidence to accept the claims that programming builds cognitive thinking. Evidence, they suggest, does exist which demonstrates social and interactional benefits of pairing children for programming activities.

In particular, these researchers refute Papert's claims that children can learn recursion (Kurland & Pea, 1983) noting in their study that 8- to 12-year-olds avoided procedures using recursion. Kurland and Pea inferred that to employ recursion the child needs to understand the terminology and logic of the language's control structures. These students, they believed, did not. Further, in a study of seven 11- to 12-year-olds in their second year of Logo programming, Kurland & Pea (1983) concluded that "self-guided discovery needs to be mediated within an instructional context" (p. 9). This conclusion was reached after allowing students to explore with Logo following Papert's "discovery" pedagogy. When

analyzing their posttest it became evident that these students had not understood the concept of recursion.

Pea (1983) concludes, after three Logo research studies, that no differences between Logo programmers and non-Logo programmers can be found because "transfer of problem solving strategies between dissimilar problems, or problems of different context, is notoriously difficult to achieve" (p. 29). His findings raise doubts about the claims made regarding Logo's ability to improve cognitive capacities. He concludes that Logo is not ineffective, since student motivation is high, but additional instruction in thinking skills and using Logo's heuristics, in a direct teaching manner may help to achieve the gains in problem solving that it claims to achieve. It appears that when a purely exploratory, discovery approach is employed in teaching children Logo, the hypothesized benefits in improved problem solving skills may be difficult to promote and assess.

Thus, this study modified Papert's pure discovery teaching style by maintaining a guided-discovery approach (see Appendix E). This study was designed to determine whether the problem solving strategies that are taught to

children in Logo programming transfer to the solving of non-routine mathematical problems.

SUMMARY AND CONCLUSIONS

In its Agenda for Action (1980) the National Council of Teachers of Mathematics recommended that problem solving be the focus of school mathematics programs. In addition, it was suggested that mathematics programs take advantage of the power of computers for children of all ages.

The primary purpose of the chapter was to review the literature related to the development and research surrounding Logo and research surrounding process problems in mathematical problem solving.

Analysis of the research related to these areas leads this investigator to conclude:

1. Educational experiences which are meaningful, non-threatening, and provide immediate reinforcement motivate children to learn.
2. Children's cognitive development matures by having multiple, varied experiences.
3. Children mature at different times and in different manners; therefore, concepts need to be

taught to them in ways which make it easy for them to understand.

4. Process problems encourage the development of problem solving strategies by focusing the child's attention on the processes used to solve problems instead of the end product of the problem.

5. Teaching children various strategies or heuristics to solve problems improves their ability to solve mathematical problems.

6. The four steps most frequently cited for successful problem solving include 1) understanding the problem, 2) developing a plan to solve the problem, 3) carrying out the plan, and 4) checking or looking back over the solution of the problem.

7. The available evidence related to the effectiveness of Logo in developing children's cognitive abilities is inconclusive. It is recommended that further research be designed with the purpose of generating substantial support for a Logo curriculum which will teach

specific heuristics to children as a supplement to their mathematics program.

The literature is replete with research on problem solving and research on Logo. However, it is an omission in mathematics education that 1) no conclusive evidence is available to support the belief that those problem solving strategies used when children program in Logo will be transferred to the solution of noncomputer problems, 2) a positive attitude exists in children who learn to program microcomputers using Logo which might generalize to a more positive attitude toward mathematics in general.

Clearly, this is a needed area of research having immediate implications for the teacher of mathematics. Thus, this study was designed to answer the following questions: 1) will children who have learned to program using Logo have a more positive attitude toward mathematics than those who have had no Logo experience, and 2) will children who have learned to program using Logo use similar strategies to solve mathematical problems without microcomputers.

CHAPTER III

Methodology

Purpose

The purpose of this study was to determine the effect of learning the programming language Logo on the problem solving strategies used by fourth grade children while they solved mathematical word problems without a computer. A secondary purpose was to investigate whether children who had learned to program with Logo would regard mathematics more favorably as measured by a mathematics attitude survey.

Sample and Sample Selection

The target population for this study was all of the fourth graders from The Park School in Brooklandville, Maryland during the 1983-1984 academic year. Specifically, the population consisted of 30 fourth grade children, 16 girls and 14 boys.

The Park School was founded by a group of people who believed that the scholastic institutions of Baltimore City did not meet their educational goals. The aim of the founders was to encourage students to

develop self-discipline and intellectual interests that would serve them through their lives.

> The school was to be non-sectarian and coeducational -- a country school in the city -- beyond pedagogical precedent. The Park School was not to tolerate force as a teaching tool. Ideally, pupils learn because they are interested, because they live their work, and because they love their school. They learn best when led by inspirational teachers who are motivated by a desire to nurture academic skills. (The Park School Handbook, 1978, p. 2)

This philosophy of education persists today, making the population of students who attend this school those whose parents believe in the philosophy as stated above.

The majority of Park School students reside within a 15 mile radius of the campus, which is located on Old Court Road, in Brooklandville, Maryland. Approximately 40 children are bussed in from Columbia, Maryland, while another 25 are bussed in from the inner city.

Most of these children come from families where education is a priority and the families' resources are sufficient to permit them to afford the tuition of an independent school. Admission tests, intelligence tests and recommendations accompany the student who seeks admission to the school. In the Lower School, students are carefully screened and interviewed by the Principal and observed by classroom teachers. Social and emotional as well as academic criteria are examined when admitting a new student. The school welcomes diversity and encourages varied points of view, values and styles, as well as ethnic and cultural heterogeneity.

> The Park School embodies in both its tradition and in its daily practice the assumptions that human beings are capable of rational self-discipline, of generosity, kindness, moral convictions and concern for others, and that learning is an expression of positive energies and fulfills natural curiosity and impulse. (The Park School Handbook, 1978, p. 3)

In August, 1983, a letter (see Appendix D) was

sent to the parents of all incoming fourth graders at The Park School. This letter indicated that from Monday, September 26th through Friday, December 16th, their children would be involved in a computer study at the school. The purpose of the study was to be explained to both the children and their parents following the termination of the study. At this same time, the stanine scores from the Mathematical Concepts section of the Educational Records Bureau (ERB) standardized tests (1982) from April, 1983 were collected for each child. A stratified sampling (Borg & Gall, 1963) was done to insure that the sample was representative of the population in terms of varying ability levels, sex, and fourth grade teacher. Children were then randomly assigned to either the experimental group or the control group. Refer to Table 1 for the breakdown of stanine scores and the grouping of children into experimental and control groups.

TABLE 1

Breakdown of Stanine Scores on the Educational Records Bureau Test (1982)

Groups Categorized by Sex	Stanine Scores								
	9	8	7	6	5	4	3	2	1
Experimental									
Male	1	0	2	0	3	1	0	0	0
Female	0	0	2	1	3	1	0	1	0
Control									
Male	0	1	2	0	2	1	0	1	0
Female	1	0	1	1	3	1	1	0	0

Instructional Unit and Materials

The experimental design used in this study was a posttest only control group design (Campbell & Stanley, 1963, p. 25). By random sampling, all factors in the internal validity were controlled, which assured lack of initial biases between groups.

During the first two weeks of the school year, children were given time to get into a general routine before they were placed in their math

groups. Once assignment was made, the control group was taught the standard fourth grade curriculum (see Appendix C) for a 45 minute period, five days each week from 10:15 a.m. until 11:00 Monday, Wednesday, and Thursday. On Tuesday and Friday the control group met for math from 8:45 a.m. until 9:30 a.m. The experimental group was also taught the standard fourth grade curriculum, but for only 20 minutes each day. During the other 25-30 minutes they used Apple II computers to learn Turtle Graphics with Logo. This group met on Monday, Wednesday and Thursday from 8:45 a.m. until 9:30 a.m. and from 10:15 a.m. until 11:00 a.m. on Tuesday and Friday.

Computer instruction was done in a laboratory where there were 11 Apple computers. One computer, in the front of the room, was attached to a large teaching screen and was used to demonstrate new commands to the children. As the weeks went by, it was also used by the children to share their projects with members of the class. Five out of the 15 children worked independently at their computers, while all other children worked in pairs. These assignments remained constant during the entire experiment.

It was determined prior to the treatment that three children in the experimental and two children in the control group had had some experience with computers the summer before in camp, but no child had worked with Logo, nor did any child have a computer in his/her home.

Children from the experimental group were told that they could use their break time, on Tuesdays and Fridays, from 11:00 a.m. until 11:20 a.m. to continue working on their Logo projects. They were also told that classroom computers (both fourth grade classrooms had Apple II computers) could be used during recess to work on projects. The experimental group was asked not to discuss the computer projects with the other mathematics class as it would spoil their instruction which they would receive after winter vacation.

Instructional Procedures

The following books were used as guides for the Logo curriculum used in this study:

Abelson, H. (1982). Apple Logo. Peterborough, New Hampshire: Byte/McGraw-Hill.

Bearden, K. M. & Muller, J. (1983). The Turtle's Sourcebook. Reston, Virginia: Reston Publishing Co., Inc.

Papert, S. (1980). Mindstorms: Children, Computers, and Powerful Ideas. New York: Basic Books.

Watt, D. (1983). Learning with Logo. New York: McGraw-Hill.

When instructing the experimental group, the traditional mathematics curriculum was taught first inside the classroom. Homework was turned in from the night before, questions about it were asked and answers given, and then the daily lesson began. After a 20 minute period the children gathered their belongings and walked to the computer laboratory where their computers had been "booted up" with Terrapin Logo. (This was done prior to their coming in to save valuable class time). Instruction, as per the Logo curriculum (see Appendix E) then began.

As part of the Logo training, students kept a journal to record their daily work. They were asked to write and illustrate each of their projects, and take notes on new commands and procedures which were introduced to them. Each child was asked to do this so that, in the event of his/her partner's absence,

work could continue.

As can be noted in the Logo curriculum, by the sixth objective children were exploring short, independent projects of their own design. From this point on, instruction, while geared to the entire class, was individualized in an effort to assist children with their projects.

The control group received a full 45 minute period on the topic of the day, with opportunities to use materials to introduce and reinforce concepts already taught. An abacus, base ten blocks and chip trading materials were used to reinforce the Hindu-Arabic numeration system, as well as regrouping and renaming in subtraction and addition. Games were developed and played to assist children with mastery of their basic facts.

As is usually the case with the standard mathematics curriculum, periodic quizzes were administered to both groups to evaluate their progress in mathematics. When given these quizzes, the experimental group was also tested on Logo material. Homework assignments were given based on the instruction given during the traditional mathematics period. These were given on a daily

basis, collected and returned to the children the following day. Consequently, the only difference between the experimental group and the control group was the 125-165 minutes per week spent using the computer to program in Logo and the increased mathematics instruction time given to the control group.

At the September 23rd parent's orientation meeting, held by the school, fourth grade parents were told, in very general terms about the computer project which would be taking place during the first trimester. They were told that all children would be taught mathematics by this experimenter during the first trimester and that one group would be receiving computer training, using Logo, while the other group would be taught only the basic mathematics curriculum. During the second trimester the group who received no computer instruction would now have this training, allowing an equal amount of time on the computers for each child. During the last trimester of the school year children would be taught the basic mathematics curriculum without further computer instruction. (It should be noted that all fourth grade children were taught to use their

classroom computers as word processors using the word processing package, Bank Street Writer. Classroom computers were used by the children as typewriters during writing periods).

Test Instruments

The pre-existing test instruments used in this study were the Educational Record Bureau (ERB) standardized test scores and the Riedesel Inventory of Children's Attitude Toward Mathematics (RICATM) survey. A four-problem problem solving posttest was developed - by the investigator - to observe the strategies used by the children when solving mathematical problems.

The ERB standardized test is a paper and pencil computer-scored test that is given by The Park School each year to its third, fourth, and fifth grade students. It tests both the aptitude and achievement of students across many areas. This test is used by independent school populations because it has been normed on both independent school populations and public school populations. In response to a request for information on this test's development and the statistical information used to demonstrate its

validity and reliability, the following information was given. The test is made up of questions that were developed using ideas from all of the independent schools that use it. A team of experts constructed the test from these questions. This yielded a test with high content validity. It was found to have a high level of construct validity, especially in regard to the Mathematics Aptitude and Mathematics Concepts portion of the test. The reliability for fourth grade Mathematics Concepts section of the test for 1982 was .8798.

It was decided that the Mathematics Concepts, rather than Mathematics Computation scores be used to group children since this test gathers data on thinking and understanding mathematics principles rather than mere ability to compute.

The RICATM was developed by Alan C. Riedesel in 1961. It uses a four choice Likert Scale to collect the feelings and attitudes that children have toward mathematics. Usually administered individually, this test was administered to the entire group. Each child had his/her own test, but items were read aloud to the group to eliminate the possibility that items would be answered incorrectly due to reading problems.

The Posttest

A four problem posttest was developed after gathering problems from Problem-mathics (1977), Aha Insight (1978), Techniques of Problem Solving, Set B (1982), and final reports of problem solving studies conducted by selected educators (e.g., Dalton, Hutcherson, Kilpatrick, Knifong, McClintock, Peacock, and Zalewski) collected by Kulm. Problems which were suitable for fourth graders and which were unfamiliar to the children at Park School were searched. In addition, it was determined that problems which could be solved using strategies similar to those used in Logo programming would be used to measure the transfer effect of Logo. These strategies included drawing a diagram, breaking a large problem into manageable parts, guessing and checking, checking over completed work and deductive reasoning. The four problems and possible solutions to them appear in Appendix F.

The investigator trained four experienced teachers to administer the posttest. Initially, these teachers received a letter (see Appendix G) which outlined the procedure and schedule for testing. The letter also noted that they were to meet and listen with the investigator to an

audiotape of the investigator administering the problem solving test to three fifth graders. The teachers were instructed by the investigator to model the procedure illustrated by the audiotapes. Further written instructions for the administration of the problem solving test were discussed (see Appendix H). Audiotaping allowed the testers to determine which thinking processes these children verbalized when solving a problem and to hear the investigator's interaction with the children.

Once the outside testers had listened to the audiotape, a discussion of the testing procedure followed. One outside tester took the four problems home and practiced on her own ten-year-old daughter to be sure that she felt comfortable administering the test.

On December 9, 1983 the control and experimental groups were told that they would be taken individually into private rooms and given four problems to work on on Monday, December 12th. They were told that the work they did on that Monday had nothing to do with their marks in mathematics, but that these problems would help the investigator with her study. The children were told to take their time

on the problems, and that if 45 minutes was insufficient they would be able to work until they completed their work. The children were also told that they would be asked to think aloud, and that they would be reminded by each of the testers to keep talking.

At 8:30 on December 12, 1983 all of the outside testers were introduced to the fourth grade children. Rooms were assigned to each of the outside testers and the investigator. Each tester had a tape recorder, blank cassettes, paper, pencils and tracing paper, in addition to the four problems which were printed on separate cards. The children were told to be prepared to leave their classroom to be tested whenever requested by a returning student. Each tester was given an equal number of experimental and control students. In addition testers were not informed which of the children comprised the experimental and control groups.

Hypotheses to be Tested

Based on related literature as well as several years of teaching experience, this investigator formulated the following hypotheses.

Hypothesis 1: Children who have learned to program using the language

Logo will use similar strategies to solve mathematical problems without microcomputers.

Hypothesis 2: Children who have learned to program using the language Logo will have a more positive attitude toward mathematics as measured by an attitude survey.

Design and Experimental Controls

A posttest only control group design (Campbell & Stanley, 1963) was used in this study. In any experiment the investigator attempts to control as many variables as possible. This investigation was designed to control the subjects' grade in school, prerequisite skills, mathematics teacher and socioeconomic background. The instructional time and setting have also been controlled. The influence of several other factors such as the subject's sex, attitude toward school, personality, exposure to previous computer experiences, and parental influence have been controlled through randomization.

Campbell and Stanley (1963) outline several variables which might threaten the validity of any experiment. The following is an explanation of the way these threats were addressed in this study.

Maturation

The two groups were rotated to different instructional periods during the week in order to avoid different effects on learning or attitude due to fatigue or hunger which could result from the specific time of instruction. The time table in Table 2 displays the times of mathematics instruction.

TABLE 2

Instructional Time Periods for Experimental and Control Groups

	8:45 - 9:30	10:15 - 11:00
Monday	Experimental	Control
Tuesday	Control	Experimental
Wednesday	Experimental	Control
Thursday	Experimental	Control
Friday	Control	Experimental

Both the experimental and control group received instruction for an 11 week period.

Instrumentation

The factors related to instructional difference were somewhat controlled since an equal number of children in the experimental and control group were from both sections of fourth grade. Further control was applied by having the investigator do all of the mathematics instruction.

A potential problem existed in that the RICATM is usually administered individually. To alleviate this threat, as much as possible, an explanation was given to each class regarding how to mark the survey before beginning to answer. In addition, the investigator read aloud each item to alleviate the threat that reading level might interfere with responding to the statement.

Mortality

Factors relating to mortality were somewhat controlled by keeping the duration of the study short, 11 weeks, and by requiring that it be a part of the regular curriculum that all must take. Periodic illnesses of students occurred in both the

control and experimental groups, and did not limit the study.

Interaction

Students in the experimental group were asked not to discuss their computer studies with the other group because it would be less exciting to them when they learned it in the winter. An honor system approach was adopted, which seemed to have worked. In order to eliminate the threat of interaction during the posttest, all children were tested during a two day period. Each of the outside testers interviewed children during these times:

8:45 - 9:30

9:45 - 10:30

10:45 - 11:30

1:00 - 1:45

2:00 - 2:45

This investigator interviewed children during the same times, as well as during the timespan of 12:00 - 12:45. Twenty-six children were tested on Monday, December 12, 1983 and the remaining four children were tested on Tuesday, December 13, 1983 by the investigator. Children were asked not to discuss the tests with any of their classmates until everyone had

taken it. On Wednesday, December 14th, all thirty children were told the purpose of the test, and the questions and possible solutions were examined.

Dependent Variable

The dependent variable, transfer of problem solving strategies, is a difficult one to measure. While talking aloud lets the researcher know what processes, strategies, or heuristics are being used (Ericsson & Simon, 1980), some researchers have expressed concern about the use of verbalizations and the thinking aloud procedures in testing situations (e.g., Kilpatrick, 1980; Nisbett & Wilson, 1977). These concerns center around questions of whether verbal performance during problem solving might interfere with the subject's performance. Another concern is whether verbal data may be incomplete since the subject may not be able to communicate each thought that occurs as he/she solves the problem. Despite these concerns, a think aloud procedure has been used by researchers who are trying to analyze the thought process of children as they solve problems (Kulm, et al, 1981; Lucas, 1978).

By providing the testers with a script to follow

during the testing, and a time period to stop student silence, the quantity of student verbalizations has been controlled. The quality of student verbalizations varied with each student and was difficult to control.

Experience of the Testers

Four outside testers conducted the problem solving posttest with this investigator and the students involved in the study.

The purpose for using multiple administrators was to hold constant the time lag between the end of instruction and the posttest.

Each outside tester was thoroughly trained on the interview techniques and was given the problems, answers, and script to insure agreement between administrators. Therefore, any error due to multiple administrators was systematic and could be controlled.

Coder Agreement

A revised process code system was used to code the processes used by the children when solving the four posttest problems (Kulm et al 1981). Two persons, this investigator and a mathematics education doctoral candidate with many years

experience in the field of mathematics education, coded the processes used by the students who solved the four word problems. In order to insure coder agreement, meetings between the coders were held to discuss the coding sequence being used and any behaviors or verbalizations which might occur in response to a code. Since coding required these investigators to listen to protocols which had been audiotaped and code each process in order of its appearance, it was important that both coders were consistent as they simultaneously coded. Lengthy discussions about the coding sequence enabled each coder to agree upon an interpretation of the symbol used to code and a trial coding of one of the 5th grade pilot study students yielded sufficient agreement to proceed with the coding of the posttest.

The string of symbols was compared for agreement using a technique for computing percentage of agreement (Campbell, 1984). The percentage of agreement for each symbol in the pre-determined listing of processes was obtained using the following formula:

$$R_i = \frac{A_i}{A_i + D_{1i} + D_{2i}}$$

R_i represented a coefficient of agreement.

A_i represented the number of times both investigators agreed that the verbalization represented by code i occurred.

D_{1i} was the number of times code i was used by coder 1 and not be 2.

D_{2i} was the number of times investigator 2 indicated code i and investigator 1 did not (Smith & Meux, 1970).

An agreement of 80% or better is considered adequate coder agreement (Campbell, 1984). Table 3 illustrates the percentage of agreement on the four pilot study problems.

TABLE 3

Percentage of Coder Agreement Prior to Coding Posttest Data

Problem Number	Percentage of Agreement
1	.8000
2	.8979
3	.9285
4a	.9166
4b	.9250

To insure continued coder agreement, every fifth audio cassette was coded simultaneously and a percentage of agreement was computed. Tables 4 – 6 illustrate the percentage of agreement on the three students' coding sequences which were coded simultaneously.

TABLE 4

Percentage of Coder Agreement Following Coding of the Ninth Student's Coding Sequence

Problem Number	Percentage of Agreement
1	.8888
2	.9067
3	.8666
4a	.8387
4b	.8947

TABLE 5

Percentage of Coder Agreement Following Coding of the Seventeenth Student's Coding Sequence

Problem Number	Percentage of Agreement
1	.8148
2	.9091
3	.8421
4a	.9167
4b	.8824

TABLE 6

Percentage of Coder Agreement Following Coding of the Twenty-Sixth Student's Coding Sequence

Problem Number	Percentage of Agreement
1	.8261
2	.8966
3	.8421
4a	.8095
4b	.8600

Statistical Analysis

The statistical analysis consisted of computation of the mean and standard deviation of the students' responses on the Riedesel Attitude Survey. A t-test was performed to compare these means. The scores from this survey can be found in Appendix K.

A Chi-Square test and a t-test were both used to analyze the coding sequences collected to test hypothesis 1. When computing the Chi-Square, Yates'

correction was used when the cell size in the two-by-two table was less than 5. Appendix J contains the coding sequences for each student.

CHAPTER IV

Results

The purpose of this experiment was to compare the use of specific problem solving strategies by fourth grade children who used Logo with those who had never been exposed to Logo. A secondary purpose was to investigate whether children who were programming with Logo would regard mathematics more favorably than those who had never programmed with Logo. The investigator taught two groups of fifteen students. Two hypotheses were generated for investigation in the study. This chapter includes the results of the statistical analysis performed on these hypotheses.

Hypothesis I: Children who have learned to program using the language Logo will use similar strategies to solve mathematical problems without microcomputers.

Two statistical tests were utilized to analyze the coding sequences generated by the testers. Five codes were selected as those representative of strategies used by children when programming in Logo. It is these five codes which are contained in Table 7. The codes

have been defined in Appendix I, but the specific codes analyzed appear below.

F – Draws a figure, table or other schematic representation

P – Plans (suggests a plan for a final or intermediate goal); states a new or different goal; suggests needed information apparently helpful in solution

C/B – Checks (checks solution to problem or subproblem); looks back (looks for alternate method, generalization, or additional conclusions); studies or reviews solution; explains answer unprompted

D_s – Deduction by synthesis (makes a deductive inference drawn from several pieces of information)

$\downarrow_*$ – Error; error corrected

Table 7 displays the number of times members of the experimental and control groups used these specific strategies. The values in Table 7 represent presence or absence of each strategy for each child rather than a cumulative count of strategy usage. (See Appendix J for individual coding sequences).

TABLE 7

Number of Students Using Each Code on Each Problem

Problems		Codes F	P	C/B	D_s	↓*
1	1*	8	10	5	12
	1**	5	5	1	6
2	1	6	9	9	11
	3	4	6	5	11
3	5	8	8	9	1
	2	4	2	9	2
4a	13	14	10	2	2
	11	15	6	1	3
4b	15	12	12	2	7
	13	13	4	4	4
Totals	35	48	49	27	33
	30	41	26	18	26

* Experimental Group
** Control Group

It can be noted that children used the strategies of planning and drawing more frequently to solve the two topology problems. The experimental group used the checking and looking back strategy more frequently than the control group to solve every problem. When comparing the totals there is a large difference between the scores on this same strategy.

To further test this data, a Chi-Square test was performed. Table 8 contains the results of this test.

TABLE 8

Chi-Square Critical Values of

Problem solving Posttest

Problem	Problem Solving Strategies				
	F	P	C/B	D_s	↓*
1	.54	1.22	3.33**	1.88	3.47**
2	.28	.15	0	2.14	.17
3	.75	1.25	3.75**	0	0
4a	.21	0	2.14	0	0
4b	.54	0	6.56*	.29	.57

* $p < .05$
** $p < .10$

The data reveal that at the .05 level (3.84) the scores were statistically significant on two different problems using the same problem solving strategy (checking and looking back). At the .10 level (2.71) the experimental group's scores were statistically significant on two additional measures.

Means and standard deviations were computed for this same data, and a subsequent t-test was performed. Table 9 displays the data.

TABLE 9

t-test on Mean Scores of Strategies Used Across All Problems

Strategy	Mean		t-value
	Experimental	Control	
F	2.33	2.00	.49
P	3.20	2.70	1.16
C/B	3.30	1.70	4.50*
D_s	1.80	1.20	1.61**
↓*	2.20	1.70	1.17

* $p < .0005$
** $p < .10$

Table 9 reveals statistical significance at the .0005 level (1.66) for the problem solving strategy of C/B (checking and looking back). Statistical significance at the .10 level (1.29) was reached on the strategy of D_s (deductive synthesis).

Based upon the results of the Chi-Square and t-tests given in Tables 8 and 9, Hypothesis 1 is partially supported. That is, students who learned to program in Logo appear to have learned certain specific strategies which they apply in solving noncomputer problems.

Hypothesis II: Children who have learned to program using the language Logo will have a more positive attitude toward mathematics as measured by an attitude survey.

Individual means and standard deviations for both groups' scores on each subsection of the Riedesel Attitude Survey were computed and reported in Table 10. A t-test was then performed to compare the mean scores of each subsection. (Individual scores are reported in Appendix K).

TABLE 10

Means and Standard Deviations and t-Test Scores of Each Subsection of Riedesel Attitude Survey

Subsection	Maximum Possible	Experimental n=15	Control n=15	t-Test
Overall Attitude	4	3.60 .51	3.40 .74	.87
General Attitude	48	40.53 4.56	37.73 5.52	1.51**
Liking for Math	36	30.33 3.99	28.73 5.56	.91
Usefulness	44	37.67 4.15	38.73 3.73	.74
Anxiety	40	16.53 5.36	17.20 6.18	.32
Interest	20	15.73 2.31	16.53 6.39	0
Ability and Achievement	44	32.33 4.03	30.00 5.20	1.37**
Understanding	8	7.27 .80	6.93 1.16	.92
Preferences	16	10.87 1.60	10.40 2.13	.37

** $P < .10$

While statistical significance at the .05 level (1.70) was not found, several interesting observations can be noted. Given the high mean scores on general attutide it is noted that children have a very favorable attitude towards school in general. It is further noted that they have low anxiety towards mathematics, as demonstrated by the low scores on the anxiety section of this test. Understanding of mathematics principles as perceived by each child is high, and interest in mathematics as a school subject is also quite high. Statistical significance at the .10 level (1.31) was noted on two subsections of the attitude survey.

It is interesting to note that the experimental mean scores were higher on 7 out of 9 of the subsections. Yet, without statistical significance, Hypothesis 2 cannot be supported.

CHAPTER V

Interpretation, Implications and Recommendations

This study was designed to investigate the effect of learning Logo on the problem solving strategies children use when solving noncomputer problems. Also investigated was the effect of learning Logo on the attitude children have toward mathematics. The findings from each investigation will be discussed below.

Hypothesis 1: Children who have learned to program using the language Logo will use similar strategies to solve mathematical problems without microcomputers.

The analysis of the codings from the problem solving posttest seem to indicate that given certain types of noncomputer problems children do utilize certain problem solving strategies which have been reinforced during Logo instruction. Specifically, the strategies of checking and looking back over one's work, as well as noticing errors and then correcting them seem to have been transferred and

used by the experimental group. No statistically significant differences exist when comparing any of the other three strategies in question. Based solely on these findings, one might infer that the time, effort and expense of incorporating Logo into a mathematics curriculum is not justified. The above findings must be considered in relation to other studies which suggest that the teaching of Logo will improve children's problem solving performance (Lawler, 1980; Papert, Watt, diSessa & Weir, 1979; Schwartz, 1984; Watt, 1979). Furthermore, the psychological and mathematical justification for teaching Logo is powerful. Brownell (1935, 1942), Bruner (1962, 1966), Papert (1980), and Piaget (1952) are just a few of the experts in education and psychology who take the position that conceptual thinking may be enhanced by providing children with meaningful educational experiences which place the student in a position of control. In addition, these same experts emphasize the important educational implications of providing students with immediate knowledge of their results in a highly motivating, nonthreatening environment. Current educators, (e.g. Abelson, 1982; Davidson, 1983; Lawler, 1980;

Solomon, 1984; Watt, 1983) suggest that Logo is a computer language which does this.

This study was designed in accord with the theoretical positions espoused by Brownell, Bruner and Piaget. Effort was made to control all variables except the teaching of Logo. Thus, the finding of partial significance in the use of problem solving strategies indicates the possibility that factors other than the treatment effect accounted for these results. Hypotheses related to phenomena which may have affected posttest results to a greater degree than the experimental treatment will be discussed below.

Problem Solving Strategies

Observation of the children during the final days of Logo instruction and observations of previous researchers testing Logo's effects (e.g. Kurland & Pea, 1983; Pea, 1983) led the investigator to infer that many children had not made a connection between Logo programming and solving mathematical problems. For example, children began new projects by immediately drawing while at the computer when they had been taught to first plan out their project by

drawing in their journals. While the strategy of checking and looking back over procedures, finding bugs and then correcting them was used frequently in the computer lab, these same students failed to check over daily noncomputer classwork and homework. The investigator frequently said, "This is a strategy that may help you solve problems," but it appears that the students defined mathematical word problems differently than they did computer problems. Further confusion over terminology and application of strategies may be a result of Park School's emphasis on daily problem solving as part of the curriculum. Children are exposed to daily classroom meetings, problem solving role-playing, discussions and writing assignments in an effort to foster responsible problem solving behavior. The expectation on the part of the faculty is that such an emphasis prepares a student to act responsibly and solve daily problems with little adult intervention. Within this context it is possible that an additional opportunity to work out problems and use specific problem solving strategies had little effect on the overall behavior of children who were already skillful in solving problems.

Another factor possibly influencing the outcome of this study was its short duration. The original Brookline studies were conducted during the entire 1977-1978 school year. Berger's (1983) observations of children at the Cotting School for Handicapped Children occurred during a period of several years. Lawler's (1980) study of his son lasted for six months. In each of these studies researchers observed a positive effect in problem solving behavior of children who programmed with Logo.

Eleven weeks is a brief time period to run a study and find significant results. Pea (1983) and Kurland and Pea (1983) found this to be true in their studies when children failed to utilize principles of recursion and variables following a year-long study. It is possible that these principles of programming were too complex for young children to learn and apply without long term use.

A similar effect may have occurred in this study. Children were taught a multitude of commands in a relatively short period of time. While most had numerous opportunities to guess and check, as well as locate "bugs" and correct them by checking over their work, few did this with repeated regularity.

Additional programming time may influence the strategies children will use.

Problem Solving in the Classroom

One additional possibility exists as to why hypothesis 1 can only be supported partially. It is likely that the additional emphasis placed on mathematical problem solving in the traditional mathematics curriculum may have impacted on the results found. Children in both groups were given varied experiences solving word problems. These ranged from textbook type problems to those found in the Techniques of Problem Solving (TOPS) series, to those found in puzzle books. Children worked in pairs as well as independently to solve these problems, and examples of the strategies were presented by the teacher during several lessons. Since the control group spent 25 minutes per day practicing the lesson that had just been presented to them, it is possible that they had sufficient time to learn and apply the same problem solving strategies that the Logo children were learning in their computer instruction time. If this was the case, then the experimental group was merely being taught

the same strategies that the control group was learning, but in a different manner.

Process Coding Sequence

For the purpose of this study Kulm's (1981) process coding sequence was used to code the strategies used by children when solving noncomputer problems. Yet, several strategies taught regularly to children learning Logo are not even included in the coding sequence. These include, 1) breaking a large problem into manageable parts, 2) guessing and checking, and 3) testing parts of a solution prior to testing the solution in its entirety. In this study the coding sequence itself may have been limited. Further investigation with a coding sequence incorporating these strategies would enable one to decide whether strategies used during Logo use were also used to solve word problems.

The coders in this study inferred that planning (P) was being done when a child said, "First I'll try all of the sevens and then I'll do all of the sixes." This kind of a statement may be a further application of deductive synthesis or even guessing and checking. It is postulated that a different

process coding sequence, perhaps one specifically designed to analyze Logo programming strategies, might be used to note the transfer effect to noncomputer word problems.

The above-mentioned explanations may have accounted for the results which were found on problem solving behavior.

Attitude

Hypothesis 2: Children who have learned to program using the language Logo will have a more positive attitude toward mathematics as measured by an attitude survey.

The results from the attitude survey notwithstanding, positive shifts in attitude in children in the experimental group were observed. Children who appeared fearful of touching the computer eagerly turned it off and on and typed in commands once instruction was underway. Children who initially panicked when they had "bugs" in their procedures appeared less nervous and less anxious when errors appeared in their programs. Children who allowed their partner to do most of the typing at the

beginning of the instructional time became more assertive and willing to share in the typing. Most importantly, all children looked forward to going down to the computer laboratory. They asked to remain after class ended. Many used their break times to work on their projects, and on rainy afternoons there was frequently an argument as to who would be allowed to use the one classroom computer to work indoors during recess. Children chatted about their projects with their classmates, told parents about what they were working on, and eagerly invited their parents to an exhibition of their work. Every child expressed sadness about having to stop the computer classes and many asked when they would be allowed to use Logo again.

This type of enthusiasm has been detailed by other researchers (Berger, 1983; Lawler, 1980; Papert, 1973; Perlman, 1974; Schwartz, 1984) and it appears to have been a part of this study as well.

One possible explanation for the lack of statistical significance between the attitudes of the experimental and control groups may be the lack of association that students felt between Logo and mathematics. Children did not refer to their

computer time as math; they referred to it as "Computers." Perhaps their attitudes towards mathematics were not more positive than the control group's attitudes because they did not associate their work with computers with their liking or dislike of mathematics.

Children in both groups demonstrated a very positive attitude about school in general, and there can be no explanation why the experimental group scored significantly higher on this measure than the control group. The positive attitude towards school can be noted in the overall demeanor of the students. Children arrive at school several days prior to its opening, every year. It's as though they are eager to have it begin again. They come to school before the school day is supposed to begin and involve themselves in classroom activities that are provided for them by the teacher. Children express sadness over the closing of school, even during vacation times, although they look forward to breaks in the school year. The overall impression that a visitor gets of the children in the school is that they are happy, enthusiastic, and interested learners. Perhaps this one experience with Logo is

insufficient to alter their attitudes. Once again a different school population might demonstrate results favoring the hypotheses of this study.

One final possibility exists regarding the reasons why the attitudes of the experimental group were not different than those of the control group. Children are motivated to please the adults around them. They want teacher approval and often know what to do to get it. Many of the children answering the attitude survey may have responded in ways they felt would please the investigator. This may have had an effect on the answers they gave.

Summary

In summary, there are several factors, other than the teaching of Logo which may have affected students' learning in this experiment. Among these are 1) the children's definition of problem solving, 2) the length of the study, 3) the process coding sequence used, 4) the effect of increased problem solving within the traditional mathematics curriculum, and 5) the overall attitude of the student population.

In general, the results of the study indicate that one is likely to find an increased use of the checking back and correcting error strategies, but that statistical differences in attitude and the use of other strategies are not likely to be found in an 11-week study with fourth grade children from a moderate-high socio-economic surburban school.

Implications

While the statistical implications favoring Logo in helping children solve problems is unconvincing, the educational implications seem great. Increased emphasis on mathematical problem solving in the mathematics curriculum provides students with practical applications for their mathematics understanding. Students profit from a varied assortment of mathematics problems and from direct teaching of problem solving strategies.

Today's children are being inundated with information about computers and the potential of computer technology. They need experiences with computers which are positive, nonthreatening, and at their developmental level. At the same time, it would be worthwhile if the computer language they

were taught gave them opportunities to think on their own, solve problems in order to program, and become comfortable around this piece of technological equipment.

Researchers and educators alike report positive effects such as increased socialization between children (Pea, 1983), increased problem solving abilities (Papert, 1980), increased motivation and desire to work on computer programming (Solomon, 1984) and increased understanding of mathematical concepts, such as variables and recursion (Seidman, 1981).

Since the areas of problem solving and computers are targeted for emphasis in the 1980's, it seems important to tie these two areas together whenever possible. Logo has this potential. Teachers may reinforce the strategies they are teaching during the mathematics period with additional emphasis on them during Logo instruction. The more children use these strategies the more likely they will be to apply them whenever they solve problems.

It would seem that any tool a teacher can use to motivate, stimulate and teach children to solve problems would be a worthwhile tool to use. The

teaching of Logo may just be the computer language that teachers can use to further reinforce problem solving strategies while familiarizing children with computer capabilities.

The language is as easy for an adult to learn as it is for a child, and college courses and teacher in-service workshops can prepare the interested teacher in a minimal amount of time. A semester long course, with plenty of hands-on-experience and appropriate readings can prepare the teacher to incorporate the teaching of Logo into the mathematics curriculum.

Recommendations

This research study has raised many questions concerning the effects of teaching Logo on both the problem solving strategies used by and the attitudes of fourth grade children as they program. This final section will address future research which is recommended.

1) This study should be replicated with an increased experimental period. A year long study would allow a teacher additional time to stress the strategies, and would give

children multiple opportunities to apply them.

2) This study should be replicated with the subjects stratified according to their ability or inability to demonstrate formal operational thinking. Thus, it would be possible to ascertain if the Logo experience is different for children who have reached the level of formal operational thinking than those who have not.

3) This study should be replicated using a different population of school children. Children from urban or rural schools, children from public schools, or children from traditional private schools need to be examined to determine whether the effects of Logo learning vary with the population being studied.

4) This study should be replicated using public school children who have limited access to computers to determine whether their attitude toward computers and/or mathematics varies with exposure to Logo.

5) This study should be replicated using a different attitude survey to determine the attitudes children have towards computers prior to and then following the teaching of Logo within the mathematics curriculum.

6) This study should be replicated using a process coding sequence which specifies Logo strategies and which gives the coders exact problem solving strategies to code.

7) This study should be replicated using either an older or younger population to determine the effects of Logo learning on children of different ages.

8) A follow-up study of the children involved in this study should be done to ascertain whether the strategies employed when solving Logo problems continue to be used in higher levels of mathematics.

BIBLIOGRAPHY

Abelson, H. (1982). A beginner's guide to Logo. Byte: The Small Systems Journal, 7, 88-112.

Abelson, H. (1982). Apple Logo. Peterborough, New Hampshire: Byte/McGraw-Hill.

An Agenda for Action: Recommendations for School Mathematics of the 1980's. (1980). Reston, Virginia: National Council of Teachers of Mathematics, Inc.

Bank Street College of Education for Children and Technology. (1982). 1981-82 Annual Report. New York: Bank Street College of Education.

Bearden, K. M. & Muller, J. (1983). The Turtle's Sourcebook. Reston, Virginia: Reston Publishing Co., Inc.

Berger, A. R. (1983). Learning from bugs. Unpublished manuscript. Massachusetts Institute of Technology, Educational Computing Group, Cambridge, Massachusetts.

Borg, W. G. & Gall, M. D. (1971). Educational research: An introduction (2nd Ed.). New York: David McKay.

Bracey, G. W. (1982). Computers in education: What the research show. Electronic Learning, 51-52, 54.

Branca, N. A. (1980). Problem solving as a goal, process, and basic skill. In S. Krulik & R. E. Reys (Eds.), Problem solving in school mathematics (1980 Yearbook) (pp. 3-8). Reston, Virginia: NCTM.

Brownell, W. A. (1935). Psychological considerations in the learning and teaching of arithmetic. In The teaching of arithmetic (Tenth Yearbook). (pp.1-31). Reston, Virginia: NCTM.

Brownell, W. A. (1942). Problem solving. In Psychology of learning (Forty-first Yearbook). (pp. 415-443). Chicago: National Society for the Study of Education.

Bruner, J. S. (1962). On knowing. Cambridge, Massachusetts: Harvard University Press.

Bruner, J. S. (1966). Toward a theory of instruction. Cambridge, Massachusetts: Harvard University Press.

Butts, T. (1980). Posing problems properly. In S. Krulik & R. E. Reys (Eds.), Problem solving in school mathematics (1980 Yearbook). (pp. 23-33). Reston, Virginia: NCTM.

Campbell, D. T. & Stanley, J. C. (1963). Experimental and quasi-experimental designs for research. Chicago: Rand McNally.

Campbell, P. F. (1984). Measuring coder agreement among coders who use a process-sequence system. Journal for Research in Mathematics Education, 15, 59-64.

Carpenter, T. R. (1980). Problem solving in mathematics: National assessment. Educational Leadership, 37, 562-563.

Carpenter, T., Coburn, T. G., Reys, R. E., Wilson, J. W. (1976). Notes from National assessment: Word problems. Arithmetic Teacher, 23(5), 389-393.

Charp, S., Bozeman, W. C., Altschuler, H., D'Orazio, R., Spuck, D. W. (1982). Layman's guide to the use of computers in education. Washington, D. C.: University of Iowa Publications Dept., Association for Educational Data Systems.

Clements, D. H. & Gullo, D. F. (1984, April). Effects of computer programming on young children's cognition. Paper presented at the annual meeting of American Educational Research Association, New Orleans.

Davidson, L.J. (1983). Thinking backward. Unpublished manuscript, The Phoenix School, Cambridge, Massachusetts.

Dewey, J. (1900). The school and society. Chicago: University of Chicage Press.

Dyrli, O.E. (1983). Logo, a language that enpowers children. Learning, 12(10), 61-68.

Ericsson, K.A. & Simon, H.A. (1979). Thinking-aloud protocols as data. Psychological Review, 87, 215-251.

Ericsson, K. A. & Simon, H. A. (1980). Verbal reports as data. Psychological Review, 87, 215-251.

Eyester, R.H. (1981). Seymour Papert and the Logo universe. Creative Computing, 7(12), 70, 72, 74.

Feinberg, L. (1983). Computers contribute little to education, study finds. The Washington Post, 285, 1, 6.

Feurzeig, W., Papert, S., Bloom, M., Grant, R., Solomon, C. (1969). Programming languages as a conceptual framework for teaching mathematics (Report No. 1899). Cambridge, Massachusetts: Bolt, Beranek & Newman, Inc.

Frick, T. & Semmel, M.I. (1978). Observer agreement and reliabilities of classroom observational measures. Review of Educational Research, 48, 157-184.

Furth, J. G. & Wach, H. (1974). Thinking goes to school: Piaget's theory in practice. New York: Oxford University Press.

Gardner, M. (1978). Aha Insight. New York City: W. H. Freeman & Co.

Garrett, H. E. (1926). Statistics in psychology and education. New York: David McKay Co., Inc.

Gawronski, J. D. (1981). Computer literacy and school mathematics. Mathematics Teacher, 74, 613-614.

Glenn, A. D. & Ellis, A. K. (1982). Direct and indirect methods of teaching problem solving to elementary school children. Social Education, 46, 134-136.

Goldenberg, E. P. (1982). Logo - A cultural glossary. Byte: The Small Systems Journals, 7(8), 210-228.

Grady, T. M. & Gawronski, J. D. (Eds.). (1983). Computers in curriculum and instruction. Alexandria, Virginia: Association for Supervision and Curriculum Development.

Greenes, C. E., Spungin, R., Dombrowski, J. M. (1977). Problem-mathics: Mathematical challenge problems with solutions. Palo Alto, California: Creative Publications.

Greenes, C., Immerzeil, G., Ockenga, E., Schulman, L., Spungin, R. (1982). Techniques of Problem Solving (TOPS), Deck B. Palo Alto, California: Creative Publications.

Harvey, B. (1982). Why Logo? Byte: The Small Systems Journal, 7(8), 163-164, 166, 170, 172, 174, 176, 178, 180, 182, 186, 188, 191, 193.

Hassett, J. (1984). Computers in the classroom. Psychology Today, 18(9), 22-28.

Hatfield, L. L. & Kieren, T. E. (1972). Computer-assisted problem solving in school mathematics. Journal for Research in Mathematics Education, 3, 99-112.

Hawkins, J. (1983, April). Learning Logo together: The social context. Paper presented at the symposium of the American Educational Research Association, Montreal, Canada.

Herbert, S.A. (1976). Learning with understanding. Columbus, Ohio: Center for Science, Mathematics, and Environmental Information. (ERIC Document Reproduction Service No. ED 113 206)

Howe, J.A.M., O'Shea, T. & Plane, F. (1981). Teaching mathematics through LOGO programming: An evaluative study. In R. Lewis & E.D. Tagg (Eds.), Computers in Education (pp. 150-151). Lausanne, Switzerland: North-Holland Publishing.

Johnson, M.L. (1983). One point of view - Mathematics equity: Computers. Arithmetic Teacher, 31(3), 2.

Judd, D.H. (1983). Programming: An experience in creating a useful program or an experience in computer control. Educational Computer, 3(2), 20-22.

Kantowski, E.L. (1974). Processes involved in mathematical problem solving. Dissertation Abstracts International, 36, 2734A. (University Microfilms No. 75-23764)

Kantowski, M.G. (1977). Processes involved in mathematical problem solving. Journal for Research in Mathematical Education, 8, 163-180.

Kilpatrick, J. (1967). Analyzing the solution of word problems in mathematics: An exploratory study. Unpublished doctoral dissertation, Stanford University.

Kilpatrick, J. (1980). Reaction paper No. 2. In G. A. Goldin & C. E. McClintock (Eds.), Task Variables in mathematical problem solving. (pp. 463-471). Columbus, Ohio: ERIC Clearinghouse for Science, Mathematics & Environmental Education.

Krulik, S. (1977). Problem solving: Some considerations. Arithmetic Teacher, 25(3), 51-52.

Krulik, S. & Rudnick, J. (1980). Problem-solving: A handbook for teachers. Boston: Allyn & Bacon, Inc.

Kulm, G., Campbell, P. F., Frank, M., Talsma, G. (1981, April). Analysis and synthesis of mathematical problem solving processes. Paper presented at the meeting of the National Council of Teachers of Mathematics, St. Louis.

Kurland, D. M. & Pea, R. D. (1983). Children's mental models of recursive Logo programs (Tech. Rep. No. 10). New York: Bank Street College of Education, Center for Children and Technology.

Lawler, R. W. (1980). Extending a powerful idea (Logo Memo No. 58). Cambridge, Massachusetts: Massachusetts Institute of Technology, Artificial Intelligence Laboratory.

Lawler, R. W. (1982). Logo ideas. Creative Computing, 8(11), 185.

Lawler, R. W. (1983). Logo ideas. Creative Computing, 9(1), 132.

LeBlanc, J. F. (1977). You can teach problem solving. Arithmetic Teacher, 25(2), 16-19.

LeBlanc, J. F., Proudfit, L., & Putt, I. J. (1980). Teaching problem solving in the elementary school. In S. Krulik & R. E. Reys (Eds.), Problem solving in school mathematics (1980 Yearbook)(pp. 104-116). Reston, Virginia: NCTM.

Lee, K. S. (1982). Fourth graders' heuristic problem-solving behavior. Journal for Research in Mathematics Education, 13, 110-123.

Lester, J. F., Jr. (1977). Ideas about problem solving: A look at some psychological research. Arithmetic Teacher, 25(2), 12-14.

Lough, T. (1983). Logo: Discovery learning with the classroom's newest pet. Electronics Learning, 2(6), 49-53.

Lucas, J. F. (1978). Induction problems and solution state-spaces. A working paper prepared by John F. Lucas in conjunction with the project Use of Heuristics in problem solving: An exploratory study (funded by NSF Grant #SED 77-18543). presented at NCTM Annual Meeting, San Diego, California.

Lucas, J. F., Branca, N., Goldberg, D., Kantowski, M. G., Kellogg, H., & Smith, J. P. (1980). A process-sequence coding system for behavioral analysis of mathematical problem solving. In G. A. Goldin & C. E. McClintock (Eds.), Task variables in mathematical problem solving (pp. 353-378). Columbus, Ohio: ERIC Clearinghouse for Science, Mathematical and Environmental Education.

Maier, N. R. F. (1970). Problem solving and creativity in individuals and groups. Belmont, California: Brooks-Cole.

Malone, T. W. (1981). Toward a theory of intrinsically motivating instruction. Cognitive Science, 4, 333-369.

Masalski, W. J. (1978). Mathematics and the active learning approach. Arithmetic Teacher, 26(1), 10-12.

Milner, S. (1973). The effects of computer programming on performance in mathematics. Paper presented at the meeting of the American Educational Research Association, New Orleans, Louisiana.

Morgan, C. (1982). Keeping our technological edge. Byte: The Small Systems Journal, 7(8), 6, 8, 10, 14, 18.

Muller, J. (1982). The friendly computer language. Creative Computing, 8(10), 55-60.

Musser, G. L. & Shaughnessy, J. M. (1980). Problem-solving strategies in school mathematics. In S. Krulik & R. E. Reys (Eds.), Problem solving in school mathematics (1980 Yearbook)(pp. 136-145). Reston, Virginia: NCTM.

Nisbett, R. E. & Wilson, T. D. (1977). Telling more than we can know: Verbal reports on mental processes. Psychological Review, 84, 231-259.

Norris, D. O. (1981). Let's put computers into the mathematics curriculum. Mathematics Teacher, 74, (24-26).

Papert, S. (1971). Teaching children thinking (Logo Memo No. 2). Cambridge, Massachusetts: Massachusetts Institute of Technology, Artificial Intelligence Laboratory.

Papert, S. (1972). Teaching children thinking. Mathematics Teaching, 58, 2-7.

Papert, S. (1973). Uses of technology to enhance education (Logo Memo No. 8). Cambridge, Massachusetts: Massachusetts Institute of Technology. Artificial Intelligence Laboratory.

Papert, S. (1980). Mindstorms: Children computers, and Powerful Ideas. New York: Basic Books.

Papert, S. (1981). Computers and computer cultures. Creative Computing, 7(3), 82-92.

Papert, S. (1982). . . . . Computers are objects to think with. Instructor, 91(7), 86-89.

Papert, S., Watt, D., DiSessa, A. & Weir, S. (1979). Final Report of the Brookline Logo project, Part II: Project Summary and Data Analysis (Logo Memo No. 53). Cambridge, Massachusetts: Massachusetts Institute of Technology, Artifical Intelligence Laboratory.

The Park School of Baltimore Handbook, (1978). Unpublished manual, The Park School, Brooklandville, Maryland.

Pea, R. D. (1982). Logo programming and problem solving (Tech. Rep. No. 12). New York: Bank Street College of Education, Center for Children and Technology.

Pea, R. D. & Kurland, D. M. (1983). On the cognitive prerequisites of learning computer programming (Tech. Rep. No. 6). New York: Bank Street College of Education, Center for Children and Technology.

Pea, R. D. & Kurland, D. M. (1984). On the cognitive effects of learning computer programming: A critical look (Tech. Rep. No. 9). New York: Bank Street College of Education, Center for Children and Technology.

Perlman, R. A. (1974). TORTIS: Toddler's own recursive turtle interpreter system (Logo Memo No. 9). Cambridge, Massachusetts: Massachusetts Institute of Technology, Artificial Intelligence Laboratory.

Piaget, J. (1951). Play, dreams, and imitation in childhood. New York: Norton.

Piaget, J. (1952). The child's conception of number. New York: William Norton & Co., Inc.

Polya, G. (1951). How to solve it: A new aspect of mathematical method (3rd ed.). Princeton, New Jersey: Princeton University Press.

Polya, G. (1980). On solving mathematical problems in high school. In S. Krulik & R. E. Reys (Eds.). Problem solving in school mathematics (1980 Yearbook)(pp. 1-2). Reston, Virginia: NCTM.

Priorities in School Mathematics: Executive Summary of the PRISM Project. (1981). Reston, Virginia: National Council of Teachers of Mathematics, Inc.

Rheingold, H. (1983). Video games go to school. Psychology Today, 17(9), 37-46.

Riordon, T. & Martin, K. (1983). The LOGO Center. The Computing Teacher, 10(8), 48-50.

Ross, S. M. & Campbell, L. (1983). Computer-based education in the Montessori classroom: A compatible mixture? Technological Horizons in Education (T.H.E. Journal), 10, 105-110.

Ross, P. & Howe, J. (1981). Teaching mathematics through programming: Ten years on. In R. Lewis & E. D. Tagg (Eds.), Computers in Education (pp. 143-148). Lausanne, Switzerland: North-Holland Publishing.

Schiffman, G., Tobin, D., Buchanan, B. (1982). Microcomputer instruction for the learning disabled. Journal of Learning Disabilities, 15, 557-559.

Schoen, H. L. & Blume, G. W. (1984), April). The effects of learning computer programming on student's performance in mathematical problem solving. Paper presented at the annual meeting of the National Council of Teachers of Mathematics, San Francisco.

Schoenfeld, A. H. (1980). Heuristics in the classroom. In S. Krulik & R. E. Reys (Eds.), Problem solving in school mathematics (1980 Yearbook)(pp. 9-22). Reston, Virginia: NCTM.

Schoenfeld, A. H. (1982). Measures of problem-solving performance and of problem-solving instruction. Journal for Research in Mathematics Education, 13, 31-49.

Schwartz, S. L. (1984, April). Microcomputers and young children: Using research to plan programs. Paper presented at the National Conference on Computers and Young Children, Newark, Delaware.

Seidman, R. H. (1981). The effects of learning a computer programming language on the logical reasoning. Paper presented at the annual meeting of the American Educational Research Association, Los Angeles, California.

Sharon, S. (1980). Learning in small groups: Recent methods and effects on achievement, attitudes, and ethic relations. Review of Educational Research, 50, 241-271.

Siegel, S. (1956). Nonparametric statistics for the behavioral sciences. New York: McGraw-Hill Book Company.

Simon, H. A. (1975). Learning with understanding. Mathematics Education Report. ERIC Information Analysis Center for Science, Mathematics, and Environmental Education, The Ohio State University.

Skypek, D. H. (1977). Problem: Solving problems. Arithmetic Teacher, 25(2), 5.

Smith, B. O. & Meux, M. O. (1970). A study of the logic of teaching. Urbana, Illinois: University of Illinois Press.

Solomon, C. (1984, April). Logo: Past and future. Paper presented at the annual meeting of the National Council of Teachers of Mathematics, San Francisco.

Solomon, C. J. & Papert, S. (1976). A case study of a young child doing turtle graphics in LOGO (Logo Memo No. 28). Cambridge, Massachusetts: Massachusetts Institute of Technology, Artificial Intelligence Laboratory.

Statz, J. (1974). The development of computer programming concepts and problem solving abilities among ten-year-olds learning Logo. Dissertation Abstracts International, 34, 5418B-5419B. (University Microfilms No. 74-10, 180).

Steffin, S. A. (1983). Fighting against convergent thinking: Using the micro as a weapon. Childhood Education, 9, 255-258.

Sturdivant, P. (1981). Introduction. In Thomas, J. L. (Ed.) Microcomputers in the schools, Phoenix, Arizona: Onyx Press.

Sudyam, M. (1980). Untangling clues from research in problem solving. In S. Krulik & R. E. Reys (Eds.), Problem solving in school mathematics (1980 Yearbook) (pp. 34-50). Reston, Virginia: NCTM.

Sudyam, M. N. & Weaver, F. J. (1977). Research on problem solving: Implications for elementary school children. Arithmetic Teacher, 25(2), 40-42.

Taylor, R. (1980). The computer in the school: Tutor, tool, tutee. New York: Teachers College Press.

Thornburg, D. D. (1982). Pilot turtle geometry. Classroom Computer News, 2(4), 38-43.

Thorton, C. A. & Bley, N. S. (1982). Problem solving: Help in the right direction for L. D. students. Arithmetic Teacher, 29(6), 26-27, 38-41.

Trafton, P. R. (1972). Individualized instruction: Developing broadened perspectives. Arithmetic Teacher, 19(1), 7-12.

Uhlig, G. E. (1982). Electronic education: Dimensions and directions. Education, 103(2), 106-111.

Wadsworth, B. J. (1971). Piaget's theory of cognitive development. New York: David McKay Co., Inc.

Watt, D. H. (1979). Final Report of the Brookline Project, Part III: Profiles of Individual Student's Work (Logo Memo No. 54). Cambridge, Massachusetts: Massachusetts Institute of Technology, Artificial Intelligence Laboratory.

Watt, D. H. (1979). A comparison of the problem solving styles of two students learning LOGO: A computer language for children. Creative Computing, 5(12), 86-89.

Watt, D. H. (1982). Logo in the schools. Byte: The Small Systems Journal, 7(8), 116-134.

Watt, D. H. (1983). Learning with Logo. New York: McGraw-Hill.

Watt, M. (1982). What is Logo? Creative Computing, 8(10), 112-119.

Webb, N. L. (1979). Processes, conceptual knowledge, and mathematical problem-solving ability. Journal for Research in Mathematics Education, 10, 83-93.

Wickelgren, W. A. (1974). How to solve problems: Elements of a theory of problems and problem solving. San Francisco: W. H. Freeman & Co.

Wills, S. (1981). Turtles for teaching computing. In R. Lewis & E. D. Tagg (Eds.), Computers in Education (p. 796). Lausanne, Switzerland: North-Holland.

Worth, J. (1982). Problem solving in the intermediate grades: Helping your students learn to solve problems. Arithmetic Teacher, 29(6), 16-19.

APPENDIX A

Example of how children use LOGO

Once children have been introduced to the computer, they are excited to begin using it. The following is a typical conversation that a teacher might have with a child who is beginning to use LOGO.

Teacher: What would you like the turtle to draw?

Student: I'd like to make it draw a shape. Maybe a box.

Teacher: All right, how shall we begin?

Student: Well, in order to make a box, the turtle first has to go up a little bit. So, I think I'll type in "GO UP".

Teacher: That means that the computer doesn't understand what you want it to do. Remember that you need to include a number that reminds the turtle how many steps to take. What other way can you think of to make the turtle move to make the first line?

Student: Well, when I walk in the shape of a box, I move forward. So, let's try "FORWARD 100".

Teacher: Good! You see, the turtle has moved and has left a trail behind it. Now what will you tell it to do?

Student: When I walk out a box, I have to turn to the right and then walk again. So let's try "RIGHT TURN".

Teacher: Again, remember that you have to include a number when you tell the computer something.

Student: (Tries a variety of numbers after the command RIGHT, and then settles on RIGHT 90 which works.) Now I need to make the turtle move again, so I'll just type in FORWARD 100.

Teacher: Good! You remembered that the first time the turtle moved it was 100, so it needs to be 100 now. What will you do now?

Student: This is easy. All I have to do is have the turtle go RIGHT 90 again, then FORWARD 100 until the box is finished.

Teacher: Why don't you try that. Now to make the turtle draw your box whenever you want it to you can type the procedure that you just developed.

Student: I'll call it "BOX". (The child now types in

TO BOX

FORWARD 100

RIGHT 90

FORWARD 100

RIGHT 90

FORWARD 100

RIGHT 90

FORWARD 100)

Teacher: Now you can make the turtle draw your box whenever you want to by just typing the command BOX.

APPENDIX B

Riedesel Inventory of Children's Attitude Toward Mathematics (RIATCM)

RIEDESEL INVENTORY OF CHILDREN'S ATTITUDE TOWARD MATHEMATICS

Child's name: _____ Date: _____

Below are some sentences that describe how people might feel about mathematics. Indicate your answers according to the following:

a. if you agree very much
b. if you agree
c. if you disagree
d. if you do not agree at all

	a	b	c	d
I. Overall Attitude Towards Math				
1. I like mathematics.	4	3	2	1
II. General Attitude Toward School				
1. I like school very much.	4	3	2	1
2. I don't like to study school subjects.	1	2	3	4
3. I have never enjoyed studying.	1	2	3	4
4. I have always enjoyed going to school.	4	3	2	1
5. There is very little need for going to school for most jobs.	1	2	3	4

		a	b	c	d
6.	Schoolwork makes me feel worried and confused.	1	2	3	4
7.	I feel relaxed and happy when doing schoolwork.	4	3	2	1
8.	School is very interesting to me.	4	3	2	1
9.	I think that school is very dull.	1	2	3	4
10.	You need to go to school in order to get a good job.	4	3	2	1
11.	I don't make very high grades in school.	1	2	3	4
12.	Most of the kids in my class know more than I do.	1	2	3	4

III. Specific Attitudes Toward Math

A. Liking for Math

1.	I like mathematics very much.	4	3	2	1
2.	It's fun to work with mathematics.	4	3	2	1
3.	I don't like to study mathematics.	1	2	3	4
4.	I have always enjoyed mathematics.	4	3	2	1

	a	b	c	d
5. I don't like mathematics very much.	1	2	3	4
6. I have a bad feeling about mathematics.	1	2	3	4
7. I have never enjoyed studying mathematics.	1	2	3	4
8. I just don't like mathematics.	1	2	3	4
9. Mathematics is one of my favorite subjects.	4	3	2	1

B. Usefulness (Utility)

1. Mathematics is very useful to everyone.	4	3	2	1
2. Mathematics is important in everyday life.	4	3	2	1
3. There is very little need for mathematics in most jobs.	1	2	3	4
4. Mathematics is not very important for most people.	1	2	3	4
5. You need mathematics in order to get a good job.	4	3	2	1
6. Mathematics is important for the country.	4	3	2	1
7. Most of what we learn in mathematics class is not useful.	1	2	3	4

		a	b	c	d
8.	Mathematics contributes to science and other fields of knowledge.	4	3	2	1
9.	Mathematics is less important in everyday life.	1	2	3	4
10.	Mathematics is a very worthwhile and necessary subject.	4	3	2	1
11.	Mathematics is less important to people than other subjects.	1	2	3	4

C. Anxiety

1.	It scares me to have to take mathematics.	4	3	2	1
2.	It makes me nervous to even think about doing mathematics.	4	3	2	1
3.	I feel relaxed and happy when working with numbers.	1	2	3	4
4.	Mathematics makes me feel worried and confused.	4	3	2	1
5.	I feel calm and confident when doing mathematics.	1	2	3	4
6.	Mathematics makes me feel nervous and uncomfortable.	4	3	2	1
7.	I feel at ease in mathematics class.	1	2	3	4

8. I am not frightened or afraid of mathematics. a b c d 1 2 3 4

9. I feel tense and tight when someone talks about mathematics. 4 3 2 1

10. I feel sure of myself when working mathematics. 1 2 3 4

D. Interest

1. Sometimes I work extra mathematics problems. 4 3 2 1

2. I would like a job that doesn't use any mathematics. 1 2 3 4

3. I am more interested in mathematics than in most other school subjects. 4 3 2 1

4. I think that mathematics is a very dull subject. 1 2 3 4

5. I find mathematics to be very boring. 1 2 3 4

E. Ability and Achievement (Perceived)

1. No matter how hard I try, I have trouble working with mathematics. 1 2 3 4

		a	b	c	d
2.	Mathematics is easy for me.	4	3	2	1
3.	I have trouble with some of the terms and symbols used in mathematics.	1	2	3	4
4.	My grades in mathematics have usually been lower than my grades in other school subjects	1	2	3	4
5.	I often think "I can't do it" when a mathematics problem seems hard.	1	2	3	4
6.	Word problems in mathematics have always been difficult for me.	1	2	3	4
7.	My mathematics grades have usually been higher than my grades in other subjects.	4	3	2	1
8.	I am good at working mathematics.	4	3	2	1
9.	I am able to work mathematics without trying very hard.	4	3	2	1
10.	I feel I could do better in mathematics if I tried harder.	4	3	2	1
11.	I remember most of the things I learn in mathematics.	4	3	2	1

		a	b	c	d
F.	Understanding				
1.	I usually understand what we are talking about in mathematics.	4	3	2	1
2.	No matter how hard I try, I cannot understand mathematics.	1	2	3	4
G.	Preferences				
1.	I'd rather work a short easy problem than a long interesting one.	1	2	3	4
2.	I like to solve new problems in mathematics.	4	3	2	1
3.	I am not willing to study mathematics any more than I have to.	4	3	2	1
4.	I would like to belong to a mathematics club.	4	3	2	1

APPENDIX C

Traditional Mathematics Curriculum

TRADITIONAL MATHEMATICS CURRICULUM

THE FOLLOWING IS AN OUTLINE OF THE TRADITIONAL MATHEMATICS CURRICULUM THAT IS TAUGHT TO ALL FOURTH GRADERS AT THE PARK SCHOOL DURING THE MONTHS OF SEPTEMBER THROUGH DECEMBER. (Adjustments are made due to general understanding of the material and these will be discussed in the results section of this paper).

I. Fourth Grade Mathematics
   - A. Numerations Systems
     1. Review of Hindu-Arabic Numeration System (Base 10)
     2. Introduction to Egyptian Numeration System
        - a. General background of symbols
        - b. Conversion of Egyptian to Hindu-Arabic and Hindu-Arabic to Egyptian
     3. Introduction of Roman Numeral System
        - a. Relationship to Egyptian and Hindu-Arabic

b. Uses of system today

4. Chinese Numeration System (introduction)

a. Explanation of type of system

b. Relationship of it to Hindu-Arabic, Egyptian, Roman

5. Development of own Numeration System (pairs)

B. Review of Addition and Subtraction

1. Introduction of Hutching's Low Stress Algorithm to Add

2. Meaning of operations

3. Computational practice

4. Solving open sentences

5. Solving simple computational word problems

C. Logic (Deductive Reasoning)

1. Problems from Mind Benders, books A_1, A_2

2. Geometrical Analogies using Attribute blocks

3. "Five Easy Pieces"

4. Problems of own making to share

D. Problem Solving Strategies

1. Choosing the operation
2. Finding facts
3. Making drawings
4. Guess and Check
5. Using Patterns
6. Making and using lists

E. Multiplication and Division

1. Meaning of the operations
2. Review of basic facts

APPENDIX D

Letter to Parents

August 15, 1983

Dear Parents,

This year we are beginning a program of computer studies for both fourth and fifth grade children. Your child will receive training in LOGO (a computer language designed for children), using one of the many Apple II computers which are now a part of our school environment.

During the first trimester, your child will be taking part (as part of the Mathematics curriculum), in a Computer and Problem-Solving study which will be conducted by Honi Bamberger as part of her research towards completion of her Ph.D. Specific information concerning the study will be provided to both you and your child following the completion of the study. During the second trimester, (after analyzing the data) a letter explaining the study will be sent to you.

We appreciate your support of this project, knowing that its results will benefit your child and the school.

Sincerely,

APPENDIX E

LOGO curriculum for fourth graders

LOGO CURRICULUM

In teaching children to program using Turtle Graphics, with LOGO, it is important to allow exploration as the means for developing programs. Initially, children will need an understanding of the basic commands of LOGO, but each will learn these at his or her own pace, and additional commands should be introduced as the individual needs them for procedures. The basic format for instruction will be a brief full group meeting for the first five minutes, where new commands are discussed and small groups are invited to share their projects. The next 20 minutes will be used for working with the computer. And a minute or two at the end of the period will be used to sum up what has been done that day and to answer any questions that have been unanswered. This format is flexible, however, and once most of the commands have been introduced, the children will probably spend most of their time using the computer and sharing their work with others. The teacher's role during this time will be to introduce new commands, help debug procedures, suggest ideas

for projects and observe the activities of each small group as they work to solve problems that arise when programming.

1 - Objectives - Introduce the four basic LOGO commands: forward, back, left, and right.

Demonstrate and teach how to turn the computer on and off and how to load the disc drive.

Explain the purpose of the RETURN key.

Explain need for numerical imputs.

Allow exploration time to see turtle's reaction to given commands.

In a full group meeting, all children will be introduced to the computer. Everyone will be instructed as to how to load the disc drive and turn the computer on and off. Each small group will be allowed to do this once groups are divided up. Divide group evenly between the computers available,

and ask the children to start up LOGO and then insert their own blank disc. Talk briefly about the keyboard and its similarities and differences to a typewriter. Explain that during the next several weeks, they will be learning commands that will help them use the computer to create interesting projects of their own design. Give the children these commands: right, left, back, and forward. Tell them that these are the basic Turtle commands and that they will be able to draw things using these commands and numerical inputs. Ask them to type in the word "DRAW." Ask, "What happened"? Introduce the turtle and then have them explore by typing in a command followed by a space and a numerical input, to see what will happen. Explain that the RETURN key tells the computer to carry out the typed command, so that for the turtle to do something, they have to press "return" after they have given a command.

Let them spend the rest of the period experimenting with the computer. At the end of the period, ask someone to summarize what they learned. Remind the children to bring in their journals to class the next day.

2 - Objectives - Review material from day 1 and share what the children discovered.

Explain how they are to keep records in their journals.

Teach the keys: ESC, ⟶ , ⟵ .

Teach the commands HOME and CLEARSCREEN (CS).

Allow exploration time for estimating distance.

Have the children get into their small groups, load the disc drive and turn on their computers. (This will be standard beginning procedure from now on). Have them open their journals and write the date and while individuals explain the four basic LOGO commands, the children should record this information into their journals. The teacher should keep a list of these and all other commands on tag board that can be kept in the front of the room. Ask if they can think of any problems that might occur when typing commands. Teach the abbreviated versions: FD, BK, RT, LT. Allow them a few minutes to practice moving the turtle around the screen with

the abbreviated commands.

Teach them what the ESC key and the → and ← key can do. Have them record this in their journals. Explain that to bring the turtle back to its original position, they must type in HOME. Have them try this. Some may notice that the turtle leaves a trace behind it, from wherever it originated. Explain that by typing in CLEARSCREEN or CS they can eliminate this line. Again have them practice using this new information and encourage them to purposely type things incorrectly so that they can use the new keys that they just learned about.

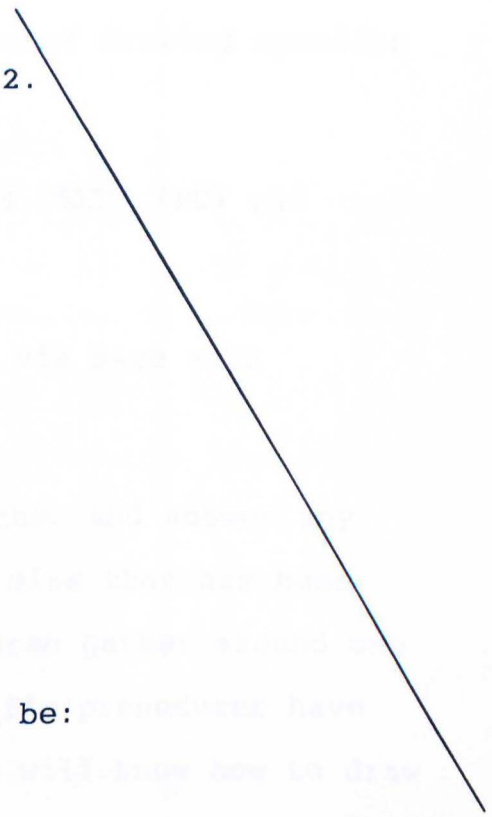
Have one child from the group point to a place on the screen and have the person at the computer try to get the turtle to reach that spot. Allow each child in each group time to try this at least twice. Then give each child the following worksheet, so that they have a chance to practice estimating "turtle lengths."

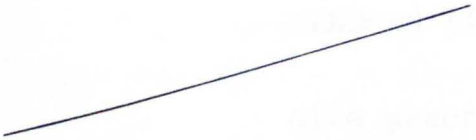
Estimating "Turtle Lengths"

If this length is 100

How long are these?

1. 

2. 

3. 

4. 

5. 

Draw a line that you think will be:

6. 400

7. 50

8. 150

End the lesson by asking for any questions and collect the assignment.

Approximate answers for the classwork assignment are:

1. 200 2. 500 3. 300 4. 50 5. 100

3 - Objectives - Introduce idea of drawing specific shapes.

Teach commands PENUP (PU) and PENDOWN (PD).

Give practice via maze with directions.

Return their classwork to them and answer any questions about it or anything else that has been covered so far. Have the children gather around one computer and explain that specific procedures have been written so that the turtle will know how to draw certain things when given one command. Demonstrate this by typing BOX, TRIANGLE, and CIRCLE. Show them how the BOX command can be used to make a fancy design by turning the turtle to the right a little before typing BOX. Then type in a shape that the computer has not been programmed to understand and ask the children to interpret what has happened.

Explain that they, as a small group have to choose a design to make, and by using the commands that they know about, teach the computer how to make their design.

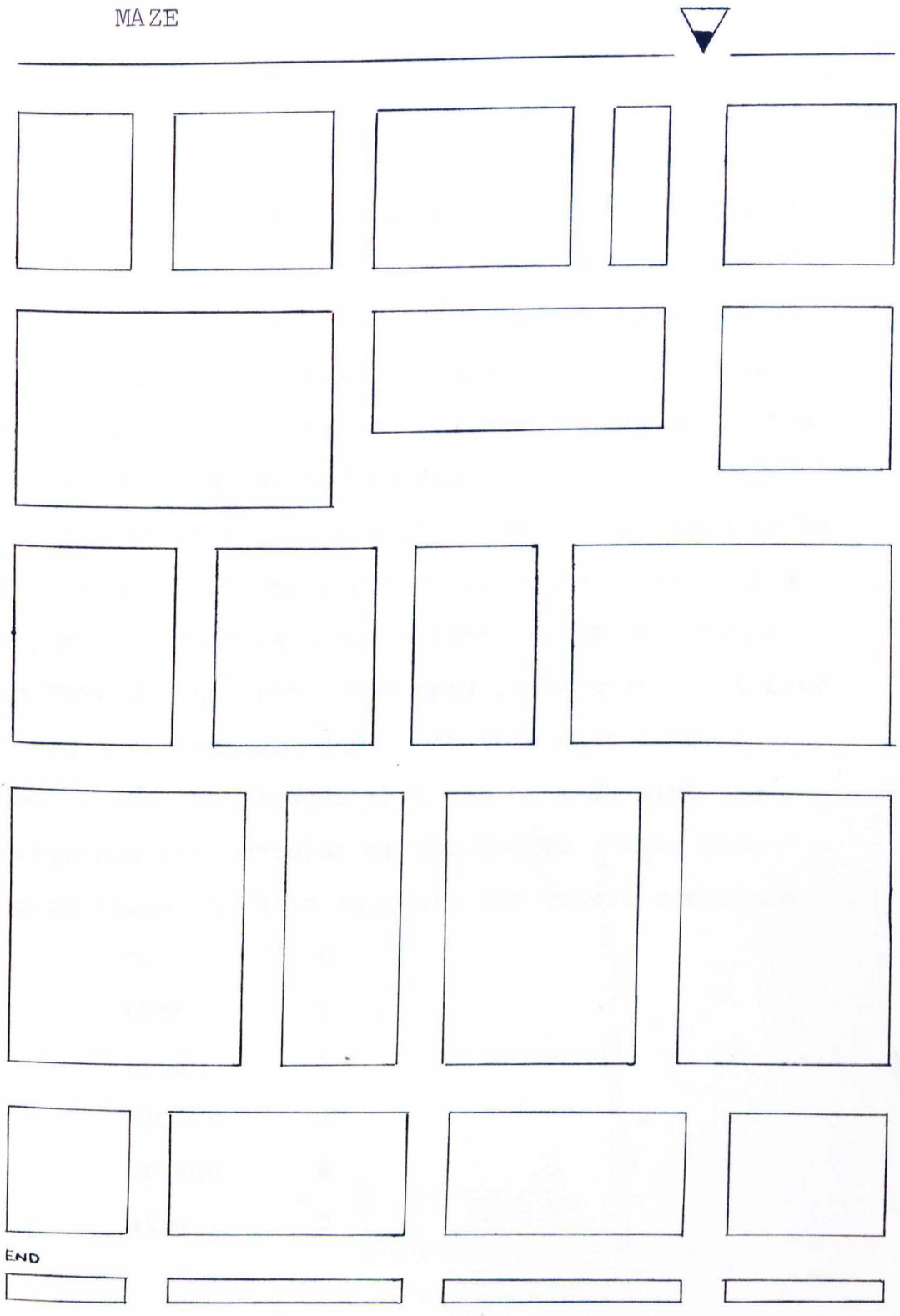
At this time teach them the commands PENUP (PU) and PENDOWN (PD) and explain that these may be useful when they make detailed pictures.

Let the children go to their computers and decide upon a simple shape that they can teach the computer to draw.

The teacher should circulate around the room and offer suggestions, provide assistance or just observe what is happening with the groups.

A few minutes before the period is to end, give the children the maze and ask them to think about how they will move the turtle from the starting point to the ending point. Tell them that they will be discussing this as a group the next day.

MAZE



4 - Objectives - Teach PENCOLOR (PC) colors.

Practice sequencing steps to reach a final goal.

Teach commands FULLSCREEN (CTRL-F);
SPLITSCREEN (CTRL-S).
TEXTSCREEN (STRL-T)

Have each child take out his/her maze and using an overhead projector overlay, map out the steps that the children say should be made to move the turtle from his start to his finish. Explain the need to be precise with the instructions that are given. At a computer, demonstrate the effect of CTRL-F, CTRL-S, and CTRL-T. Ask them when they think they might have a need for these commands. At this same time, demonstrate the changes that can be made with the background and pencolor of the screen. Have them record these in their journals for future reference.

black	0
white	1
green	2
violet	3
orange	4
blue	5

Once again, let them use the remainder of the period experimenting with the colors that they have just learned to use and forming simple designs.

5 - Objectives - Reinforce PENCOLOR (PC) and BACKGROUND (BG) color.

Introduce command REPEAT, [ ].

Experiment with circles and polygons.

On each disc the teacher has written a procedure to create circles and arcs following Watt's (1983) format:

```
TO RCIRCLE: R
REPEAT 36 [RCP: R]
END
TO RARC: R
REPEAT 9 [RCP: R]
END
TO RCP: R
RT 5
FD: R *(3.14159)/18
END
```

```
TO LCIRCLE: R
REPEAT 36 [LCP: R]
END
TO LARC: R
REPEAT 9 [LCP: R]
END
TO LCP: R
LT 5
FD: R *(3.14159)/18
END
```

Tell the children that procedures for drawing circles have been programmed into the computer, but

ask them, as a group, how they think they could get the turtle to draw a circle. Let them try to solve this by walking through a circle on the floor, or just brainstorming a solution. Explain that there are a variety of ways to do this. Use the computer to work through some of the solutions that are given. Then show this formula:

```
TO CIRCLE
REPEAT 360 [RT 1 FD 1]
END
```

Explain the use of the word REPEAT and the symbols that must envelop the section that is to be repeated. At this time ask them to redefine some of the shapes that they have already defined. Have them use the repeat symbols to do this. For example, they might redefine a SQUARE as:

```
TO SQUARE
REPEAT 4 [RT 90 FD 70]
END
TO TRIANGLE
REPEAT 3 [FD 60 RT 120]
END
```

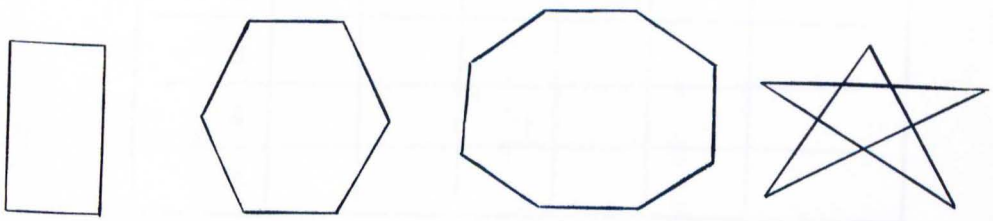
Let them know that by using REPEAT they save themselves typing errors which may create bugs in their procedures.

Let them return to their groups and experiment with pencolor changes and repeats.

6 - Objectives - Reinforce material taught.

Explore different short term projects.

This is a day to practice making shapes already made, or to try to create new shapes based on concepts that have been learned. The teacher tapes a piece of paper to the front of the room displaying these shapes:



The importance of drawing a shape on paper is discussed as an important strategy to use before beginning a project. It is referred to as planning a project. Children are told that they may try to make one of these shapes, letters of the alphabet or other simple closed shapes. They then return to their

groups for a period of exploration. Five minutes are left at the end of the period to share what different groups have been doing.

7 - Objectives - Exploration and charting of pencolor activities.

Children get into their groups and are given this ditto:

Color Chart

Background Color

PC	0	1	2	3	4	5
0						
1						
2						
3						
4						
5						

They are told to systematically explore each background color and the pencolors on it. To do this completely, they must make a list or chart to keep a record of their work. Explain that this is just another useful strategy when solving a problem. Tell them that they may make squares of different sizes

and shapes of different colors. They may keep the TURTLE at different parts of the screen or bring him home and clear the screen after each trial. Have one member of the group record the data, while another types in the commands on the computer. All members of the group should brainstorm possible ideas for completing the chart.

On the chalkboard write out this procedure:

```
BG 1
PC 0
FD 50 RT 90
PC 2
FD 50 RT 90
PC 3
FD 50 RT 90
PC 4
FD 50 RT 90
```

If they have time, they can type in this procedure and see what the TURTLE does.

8 - Objectives - Exploration of circle drawings.

Reinforcement of CTRL-F, CTRL-S, CTRL-T.

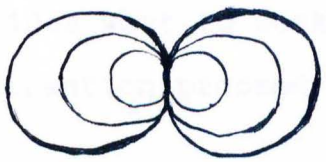
Introduce HIDETURTLE (HT);

SHOWTURTLE (ST).

Use ARCS.

While at their computers, have children type in READ "CIRCLES. Explain that by typing in the word LCIRCLE or RCIRCLE with an input number, the computer will make circles of different sizes. Let them try this for about 5 minutes.

On the board draw these things and ask them to work together to copy them onto their monitor. Remind them to keep a record of what they draw. Tell them that they may use different colors.



During the last few minutes of the period have them take out their journals and record the information the teacher gives about the commands HT and ST. Ask volunteers to explain the commands CTRL-F, CTRL-S, CTRL-F.

9 - Objectives - Review HT and ST; let children explore.

Demonstrate ARCS.

Reinforce the commands by drawing this procedure:

```
TO HEXAGON
REPEAT 6 [FD 50 RT 60]
END
```

Use the different commands during this procedure to show their effect. Then explain that like the circle procedure, the procedure for drawing arcs was programmed into the computer. Type READ "CIRCLES, and ask someone to give an input for an arc. The teacher types this in and demonstrates what happens when forming arcs. Give the groups the rest of the period to explore on the computer. Give direction to those who have no idea what to work on, while letting those who have a direction proceed uninterrupted.

10 - Objectives - Explain "teaching the computer."

Introduce Edit mode.

Review ESC and → ← keys.

Introduce CTRL-C and CTRL-G.

Work through a BUG.

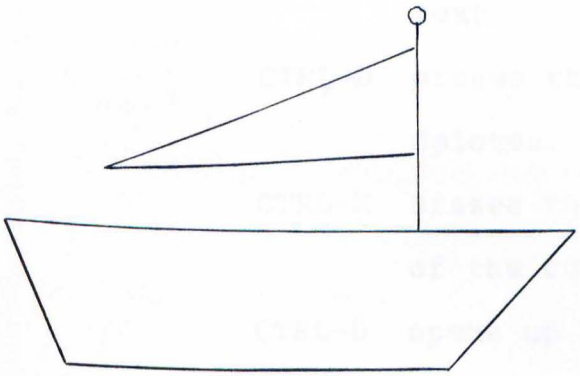
As a full group, ask how one goes about making a shape. Once this is done, explain that the computer

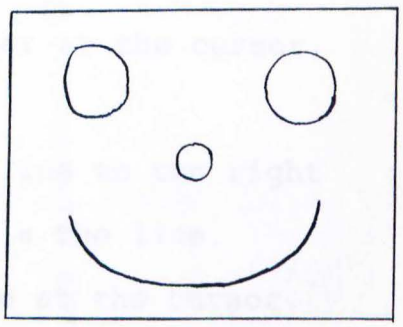
can do this on its own if it is taught how to do it. To teach the computer we type the words TO and then the name of what we wish to call our project. Tell them to take not of the difference in the monitor. Explain that this is the edit mode and that if you make a mistake in typing or in the procedure, changes can be made by getting into this mode.

Give an example of an incorrectly typed procedure and return to the editor and correct the mistake.

Demonstrate what happens when you define a procedure and then just type in the word. Explain that the procedure is lost once the machine is turned off, so they must record all of their procedures in their journal, until they learn how to save their work.

Have them try to make one of these, or begin work on a short project. Some ideas follow:





During the next nine weeks, children in small groups will plan and execute projects. They may proceed in any fashion, but as needed, small group instruction will be given to help facilitate programming. Instruction will be based on individual need and desire and will not be imposed on the learner.

Children who do a lot of editing will be given these new commands for moving the cursor:

CTRL-E moves the cursor to the end of the line.

CTRL-A moves the cursor to the beginning of the line.

CTRL-F moves the cursor forward an entire screen when the screen is full of text.

CTRL-B moves the cursor backward an entire screen when the screen is full of text.

CTRL-D erases the character at the cursor, deletes.

CTRL-K erases the entire line to the right of the cursor, kills the line.

CTRL-O opens up a new line at the cursor.

Children should record this information in their journals and then type in a procedure and test each of these commands to see how they work.

A very important concept for children to learn is how to save the work they have developed. They will have a written word in their journals, but they need to learn how to save their procedures on their discs. To do this they need an initialized disc. Once they have developed a procedure and given it a name, if they wish to save this they type in the word SAVE "name of the procedure. Children should be taught how to use the quotation marks, and be instructed to use them only before the name of the procedure, not after. Some disc drive action will occur and then the procedure is saved. The next time the child uses his disc, he need only type in the command READ "name of the procedure, and the computer will bring the procedure up from the file into the computer's working memory. This is important since children may wish to use procedures that they have developed to create more complex procedures.

Children who are having a difficult time organizing their work to do projects should be helped with these steps:

1. Draw an idea in your journal.
2. Separate this idea into parts.
3. Name each part and decide on how they will fit together.
4. Write the procedure or experiment with the computer.

For some children, this "top-down" approach is a helpful way for them to think of writing a program.

Another important and useful concept to teach children is that of the variable. Children who are ready to accept instruction in this area may be told that by using variables their projects may be easier to plan. The word variable is said to mean some information that can vary or change. Show children this example:

```
TO SQUARE  :SIZE
REPEAT 4 [FD  :SIZE RT 90]
END
```

Explain that there is no space between the dots (colon) and the variable. The procedure that was just written will make a square of varying sizes, merely by changing the imput. So for a small square, the child needs to type SQUARE 20, while for a larger one SQUARE 80 will work. Let children know that more

than just the size can vary. Explain that angles can vary, and if the procedure requires it, both can vary at once. Use a computer to write this procedure, which will vary both the length and width of a rectangle.

```
TO RECTANGLE  :LENGTH  :WIDTH
      FD  :LENGTH
      RT  90
      FD  :WIDTH
      RT  90
      FD  :LENGTH
      RT  90
      FD  :WIDTH
      RT  90
      END
```

To get a rectangle the child will need to type in RECTANGLE and then two inputs, example: RECTANGLE 60 20.

The POLY procedure is fun and exciting for many children. It is also a way for them to consolidate their procedures, and practice using variables. It should be explained that a POLY is short for the word polygon. This should be defined by them and several

examples of different polygons should be given. The teacher then gives them this procedure:

```
TO POLY  :SIZE  :ANGLE
     FD  :SIZE
     RT  :ANGLE
     POLY  :SIZE  :ANGLE
     END
```

Ask them how they will get the procedure to stop. Be sure that they know that CTRA -G will stop the procedure. Then let them return to their own computer and make different polygons. Advise them to keep a record of what they have tried. Perhaps a drawing in their journal might be worthwhile, so they remember what pair of numbers produced what polygon. If they choose to, they might try varying the angle while keeping the size the same, and then varying the size while keeping the angle the same.

After they have experimented with POLY, they need to be taught how to stop the procedure without using CTRL -G. Have them return to their own computers and

bring into the editor the procedure for POLY. Then using CTRL -O they need to open up a space after the third line in the original procedure. Once this has been done, brainstorm with them about ways they might stop the procedure. Try some of these ways, and then show them:

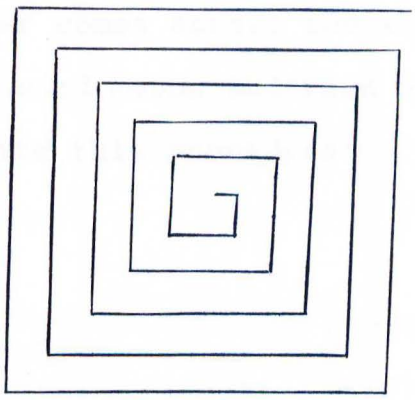
```
TO POLY  :SIZE  :ANGLE
     FD  :SIZE
     RT  :ANGLE
     IF HEADING = 0 STOP
     POLY  :SIZE  :ANGLE
     END
```

When the heading is 0 the turtle is in the home position. For those who enjoy explorations with POLY, the procedure for polyspirals can be taught. Show them an example of a polyspiral, and explain that it is a poly that grows. It either increases or decreases each time the procedure is repeated. Ask them how they might write that idea into a procedure. Then show them:

```
TO POLYSPI  :SIZE  :ANGLE
     FD  :SIZE
     RT  :ANGLE
     POLYSPI  (  :SIZE + 1)  :ANGLE
```

END

Once this procedure has been typed, show them that POLYSPI 90 makes this:



Lesson on recursion taken from:

Riordon, T. & Martin, K. (1983). The LOGOCenter. The Computing Teacher, 10(8), 48-50.

The word recursion is written on the board and using an overhead this is written:

```
TO POW
RAISE ARM
LOWER ARM
SAY POW
END
```

Remind the children that whenever they hear the word POW, they must do 3 things. Do this with them by saying "POW." Explain that whenever a word is taught to the computer, it gets stored in its memory. So when the computer comes across the word it will check to see what the word means and then do it. Next, on the overhead write this procedure:

```
TO LIVE
BREATHE IN
BREATHE OUT
LIVE
END
```

Tell them to do what is pointed to. As live is pointed to say, "The computer checks to see if there is a procedure called LIVE. Yes, there's a procedure (point to TO LIVE and here's what it means.

Have them look at

```
TO BOX
FD  50
LT  90
BOX
END
```

Run this procedure in the computer. Then teach them to BOX using variables.

```
TO BOX  :SIZE                    TO BOX  :SIZE
     FD  :SIZE                        FD  :SIZE
     RT  90                           RT  90
     BOX  :SIZE                       BOX  :SIZE + 5
     END                              END
```

ONCE THE CHILDREN ARE FAMILIAR WITH THE IDEA OF RECURSION LET THEM EXPLORE. Instruct them to develop procedures with recursion, and test them out.

APPENDIX F

Problem Solving Posttest

PROBLEM SOLVING POSTTEST

1. SUSAN WANTED TO BUY A CANDY BAR THAT COST 25¢. THE MACHINE WOULD TAKE PENNIES, NICKELS AND DIMES IN ANY COMBINATION. LIST THE DIFFERENT COINS SHE COULD USE TO PAY FOR HER CANDY.

Before beginning this problem the child will be asked to answer these questions:

a. How much does the candy bar cost?

b. What coins does the machine take?

c. Can Susan pay with a quarter?

d. What is the problem asking you to find?

By answering these questions the interviewer can eliminate the possibility that the child did not understand the problem.

There are several strategies that can be used to solve this word problem. One plan might be to make a table, similar to the one that follows.

NICKELS	PENNIES	DIMES
5	0	0
0	25	0
0	5	2
4	5	0
3	0	1
2	5	1
1	10	1
1	20	0
1	0	2
2	15	0
3	10	0
0	15	1

The child may also make a list of the combinations, or use other strategies to come up with a solution.

Upon completion of the problem, the interviewer will ask: "How do you know you have all of the possible ways"?

2. 

IF EACH SHAPE IN THE ABOVE NUMBER SENTENCE IS REPLACED BY ONE NUMBER FROM 1 - 9, HOW MANY

DIFFERENT, TRUE NUMBER SENTENCES CAN YOU MAKE? YOU MAY USE EACH NUMBER ONLY ONE TIME IN EACH NUMBER SENTENCE.

Before beginning this problem the child will be asked to answer these questions.

a. What numbers can be used in place of the shapes?

b. Can a number be used more than one time?

c. What does the answer that you will get have to be?

By answering these questions the interviewer will eliminate the possibility that the child did not understand the problem.

There are several strategies that can be used to solve this word problem. The child might use logical reasoning, a list or a guess and check method in order to solve it. One plan might be to make a list, similar to the one that follows.

ANSWER:

8 + 2 − 1	8 + 3 − 2	8 + 4 − 3
7 + 3 − 1	7 + 4 − 2	7 + 5 − 3
6 + 4 − 1	6 + 5 − 2	5 + 7 − 3
4 + 6 − 1	5 + 6 − 2	4 + 8 − 3
3 + 7 − 1	4 + 7 − 2	
2 + 8 − 1	3 + 8 − 2	

8 + 5 − 4	8 + 6 − 5	8 + 7 − 6
7 + 6 − 4	6 + 8 − 5	7 + 8 − 6
6 + 7 − 4		
5 + 8 − 4		

Upon completion of the problem, the interviewer will ask: "How do you know you have all of the possible ways"?

3. A FIREMAN STOOD ON THE MIDDLE RUNG OF A LADDER, DIRECTING WATER INTO A BURNING BUILDING. AS THE SMOKE LESSENED, HE STEPPED UP 3 RUNGS. A SUDDEN FLARE-UP FORCED HIM TO GO DOWN 5 RUNGS. LATER HE CLIMBED UP 7 RUNGS AND WORKED THERE UNTIL THE FIRE WAS OUT. THEN HE CLIMBED THE REMAINING 6 RUNGS TO THE TOP OF THE LADDER AND ENTERED THE BUILDING. HOW MANY RUNGS DID THE WHOLE LADDER HAVE?

Before beginning this problem the child will be asked to answer these questions:

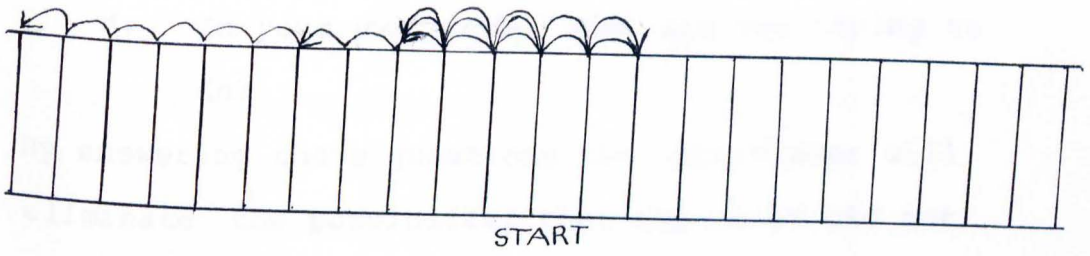
a. Where was the man standing when this problem began?

b. What are you trying to solve?

By answering these questions the interviewer will eliminate the possibility that the child did not understand the problem.

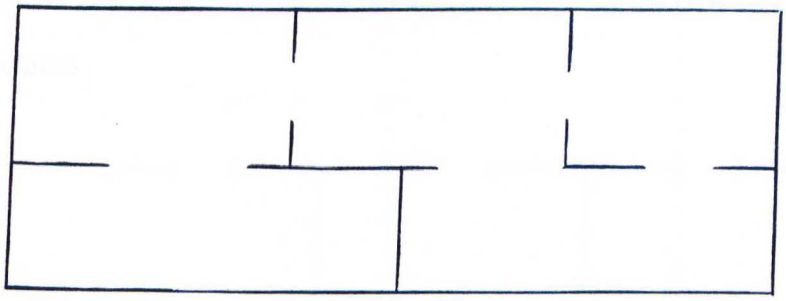
There are several strategies that can be used to solve this word problem. The child might use deductive logic, draw a diagram, develop a formula, or guess and then check his response. One plan might be to develop a drawing like the one below.

ANSWER: 23 rungs



4. CAN YOU FIND A PATH THROUGH THIS HOUSE THAT WILL TAKE YOU INTO EACH ROOM BY PASSING THROUGH EACH

DOOR ONLY ONCE? YOU MAY ENTER A ROOM MORE THAN ONCE, BUT NOT BY THE SAME DOOR.



Before beginning this problem, the child will be asked to answer these questions:

a. How many times are you allowed to pass through a door?

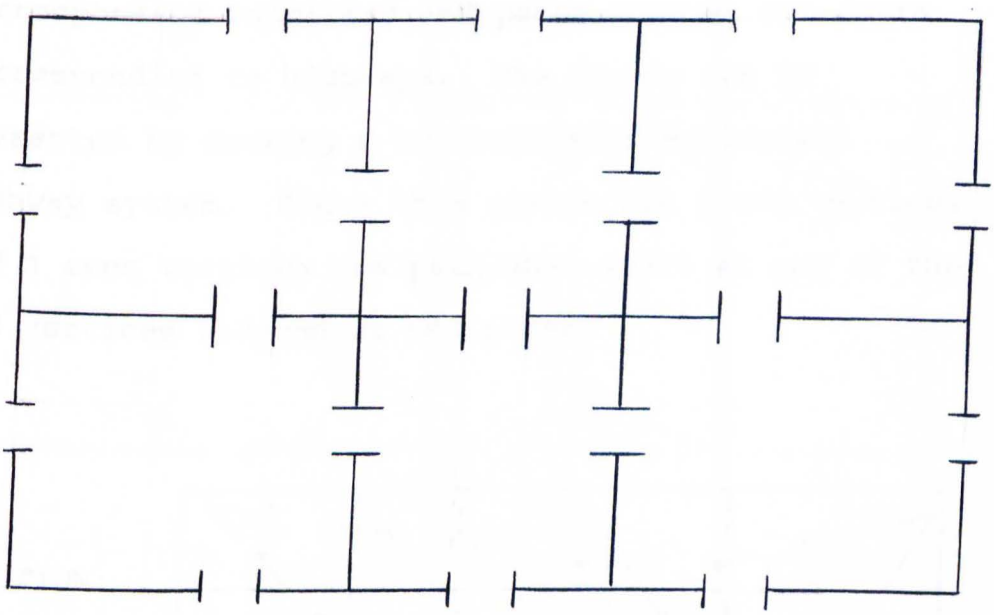
b. May you enter a room more than once?

c. If you enter the same room, may you use the same door?

d. In your own words, what are you trying to do?

By answering these questions the interviewer will eliminate the possibility that the child did not understand the problem.

BEGINNING ON THE OUTSIDE OF THIS HOUSE, CAN YOU DRAW A PATH THAT WILL TAKE YOU THROUGH EACH DOOR ONLY ONCE? YOU MUST FINISH ON THE OUTSIDE OF THE HOUSE.



Before beginning this problem the child will be asked to answer these questions:

a. Where must you begin in order to solve this problem?

b. How many times may you pass through each door?

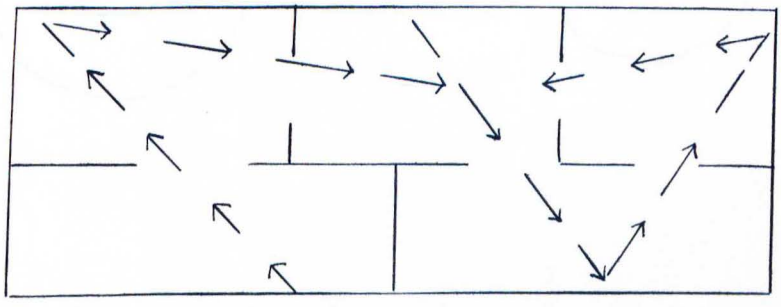
c. Where must you finish in order to solve this problem?

d. In your own words, what are you trying to do?

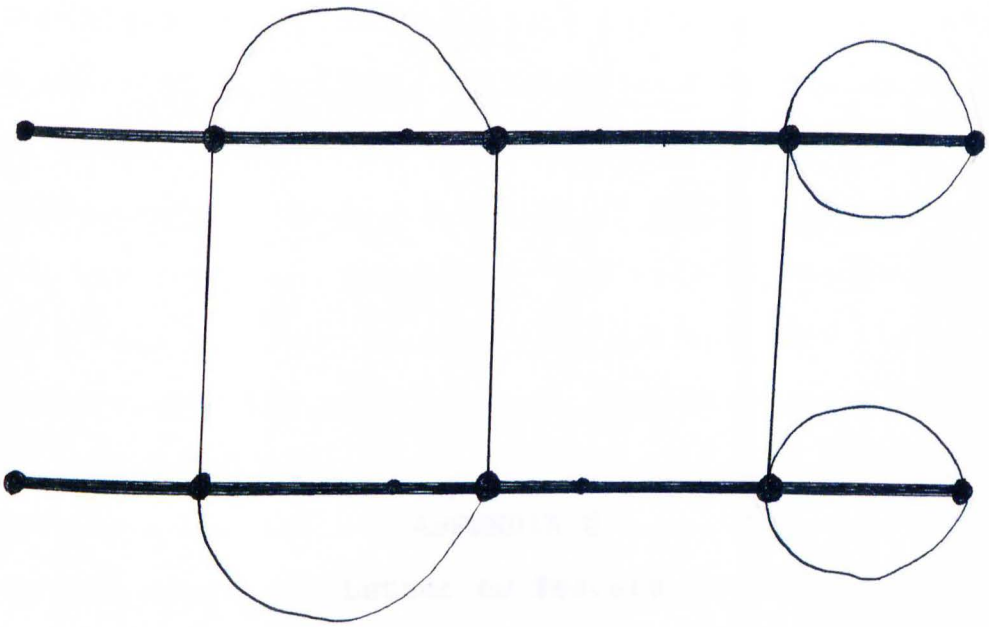
Discussion of the first part of problem 4 follows.

This is a highway problem, with the rooms corresponding to cities and paths through the doors corresponding to highways. The answer can be presented by drawing a topologically equivalent highway system. Since this system has 2 odd vertices and 3 even vertices the path must start at one of the odd vertices and end at the other.

SOLUTION:



The diagram shown to represent the second part of problem 4 looks like this:



This system has only 2 odd vertices, so it is possible to enter the house at one door and exit through the adjacent door.

APPENDIX G

Letter to Testers

December 5, 1983

Dear

The following is an explanation of what you will need to do on Monday, December 12th, in order to administer the problem-solving posttest to the fourth graders at Park School. The children should be taken to the assigned rooms (those will be given to you on Monday morning when you arrive). Before turning on the tape recorder explain to the child that the test that will be given is a puzzle type test and that they can use any strategy that they need to solve it. They may make charts, pictures, logic grids or use any other thinking strategy in order to solve the problem completely. Tell them that one very important element to the test is that they need to talk out loud during the entire test. They need to do this so that you can hear what they are thinking, even when they are not writing anything down on paper. Tell them that you will help them remember to do this by periodically reminding them to talk aloud. Give them white paper, pencils, a ruler and tracing paper and tell them that they may use any of these things to solve the problems. Let them know that there are four problems. Tell them that you

will read the problem aloud to them and they should read it along silently while you do. Let them know that you will ask them a few questions before the test begins to be sure that they understand what it asks them to do.

Then, turn on the tape recorder, hand them the first question, which will be written on a card, and read it aloud slowly to them. Then ask them the questions which appear beneath the problem. The child may look at the card to answer the questions and may refer back to the card at any time while solving the problem.

While the child solves the problem if more than 10 seconds passes when nothing is said, say either: "Please, tell me what you are thinking," or "Remember to talk out loud so I can hear what you are thinking about" or "Please tell me what you'd like to do." Vary these phrases if the child is silent for lengths of time.

If the child has no idea how to begin, and this is obvious either by verbalizations or lack of work, say, "What do you think you can do to solve this problem"? or "Is there any way that you can think of to begin solving this problem?" You may also say,

"Try to think of problems similar to this that you may have solved before--this may help you figure out what to do." Once the child is working on the problem, you may do several things to encourage his/her work:

1. Smile
2. Say, "good" or "fine"
3. Nod your head

If the child is doing something obviously wrong, remind the child to keep in mind the things that he/she is trying to solve.

If during the solving of the problem there is a pause of 15 seconds and the child doesn't seem to know what to do, you may say, "What would you like to do now"? If the child says, "I don't know," you may say, "Would you like to work on this more or go on to the next problem"? If the child says he/she wants to go to the next problem, then say, "Fine" and go on to the next problem. If the child says he/she wants to continue with the same problem, but doesn't know what to do, wait 15 seconds and then ask again, "What do you want to do now"?

Proceed with each of the four questions, having the child answer them as well as he/she can. When a

problem is completed, the child is to be asked, "How do you know that you have all of the possible answers"? (This applies to problems 1 and 2).

After 45 minutes (the tape will run out) thank the child and tell him/her that he should not discuss the test with anyone in the class until Wednesday when everyone has finished taking the test. If the child has not completed the test in the 45 minute period, turn the tape over and give additional time to complete it. Then walk the child back to the classroom and get your next child.

The schedule for Monday is:

8:45 - 9:30

9:45 - 10:30

10:45 - 11:30

1:00 - 1:45

2:00 - 2:45

I will call you to set up a time when you can listen to a practice session on a cassette that I have recorded. Thanks so much for your incredibly generous offer to help.

Sincerely,

APPENDIX H

Interview Procedure for Posttest

INTERVIEW PROCEDURE FOR POSTTEST

Students will be randomly selected from the group to be administered the posttest by a trained interviewer. Students will be tested individually and it will not be designated whether the child has been in the control or experimental group. All interviewers will be introduced to all students by the investigator who will explain to the group that they will be given individual problems to solve by one of the testers.

After an initial, brief period where the interviewer will talk to the student so that both are comfortable, a card displaying the first problem will be shown to the child. The tester will then say, "Read this problem silently with me as I read you the first problem."

Once this has been done, the interviewer will say, "Here are pencils, paper, tracing paper, and a ruler. You may use whatever you like to come up with a solution to this problem. Please talk out loud while you solve it, so that I may hear how you are thinking."

During the administration of the test, the

interviewer will be allowed to offer the student reassurances by doing the following:

a. smiling or nodding of the head

b. repeating what the student has said

c. saying, "Fine," "Okay," or "All right."

The interviewer will be allowed to repeat questions but not to rephrase them. For each of the four problems, the tester will have a sheet which has the problem that is being worked on and a set of questions that should be asked of the student before the child actually begins working on the problem. This will assure the interviewer that the child understands the problem.

If a student makes no attempt to respond after 15 seconds, the interviewer may repeat the question. If the student still does not respond after 15 seconds, the tester will then ask, "Would you like to go on"? If the student says, "No," the interviewer should wait 15 seconds and repeat the above procedure. If the student says, "Yes," the interviewer should proceed to the next item.

Prior to reading the first word problem, the interviewer will turn on the tape recorder and place it out of the way of the student. The interviewer

will be sitting to the left of the student.

If the student hesitates for longer than 15 seconds, after completing a step, the interviewer may ask, "What do you think you should do now"? If the child says, "I don't know," the tester may say, "Think about other problems that you have solved and try to see if this one looks anything like any problem you have worked on in the past." If the child continues to do or say nothing for 15 additional seconds, the investigator may say, "Why don't you try one of the strategies that you've learned this year to help you solve it." If the child still does nothing for 15 seconds, the interviewer may say, "Would you like to go on to the next problem'? If the child says, "No," the interviewer should wait 15 seconds and then repeat the above procedure. If the student says, "Yes," the tester should proceed to the next question.

After the child has completed the problem, the interviewer should ask, "How do you know that you have all of the possible ways"? If the child says, "I don't know," the interviewer might say, "What can you do to be sure that you have all of the possible solutions"?

This procedure will be implemented for each of the four word problems. After the student has completed the test, the tester will thank him/her for working with her and bring the child back to the classroom. The tester will then get the next student and give the posttest to him/her.

If during the testing a student asks if he/she is correct, the interviewer should respond, "There are no right or wrong answers. I just want to know how you think and how you will try to solve this problem." Similarly, after the testing, if a student asks how he/she did, the tester should respond, "You did just fine."

APPENDIX I

Revised Process Code Dictionary

REVISED PROCESS CODE DICTIONARY (4/1/81)

The following process code was taken from a paper that was presented at the Annual Meeting of the National Council of Teachers of Mathematics, St. Louis, on April 24, 1981. The code, itself, was developed by Dr. Gerald Kulm, Dr. Patricia F. Campbell, Martha Frank, and Gary Talsma from Purdue University.

R Reads the problem.

S Separates (indicates understanding, identifies wanted, given, relevant information).

Q Rereads the problem.

F Draws a figure, table, or other schematic representation

P Plans (suggests a plan for a final or intermediate goal; states a new or different goal; suggests needed information apparently helpful in the solution).

U Useful facts (explicitly states a concept, theorem, principle, property, or definition).

A Algorithm (uses a procedure, algorithm).

D_s Deduction by synthesis (makes a deductive inference drawn from several pieces of

information).

Z Synthesizes (summarizes information; recap of results and intermediate solutions thus far).

I Induction (states an inductive generalization or conclusion; conjecture).

Y Bright idea (proposes a solution, method, principle, in an insightful and sudden manner; change of Gestalt without apparent reason).

T Trial (conjectured or arbitrarily assigned value to a variable or to a set of variables without intervening processes).

V Variable (introduces variable or other notation).

E Equation (writes or expresses an equation).

L Like situation (reasons by analogy; perceives related problem, problem type, method, solution).

C Checks (checks solution to problem or subproblem).

B Looks back (looks for alternate method, generalization, or additional conclusions; studies or reviews solution, explains answer unprompted).

X Forgets (forgets method, concept, idea).

Outcome Symbols

1 Abandons process (followed by further attempts to solve the problem).

2 Impasse (not followed by a further attempt to solve the problem).

3 Correct final result.

4 Correct intermediate result.

5 Incorrect result.

Question Symbols

? Subject asks question.

¿ Investigator asks leading question.

~. Subject questions self.

Error Symbols

↓ Structural error.

↓ Executive error.

✳ Error corrected.

Punctuation

/ End; without solution.

. End; with solution.

APPENDIX J

Codings from Problem Solving Posttest

E_1

1. _ ? _ S _↓✳ _ S _ S _ S A _↓✳ A↓✳A C↓✳ A↓✳ A↓✳A↓✳A↓✳C A X A↓✳A↓ _D_S 2 /

2. _ S _ S ? _ S ? _↓ċ S✳A↓ċ✳↓✳ _ A E A ? _ S C A↓ ? A↓ _ S A ↓✳X A↓↓✳↓D_S↓ ↓ ↓ A E X ↓ A U B U↓ X A↓ B A X ↓✳B 2 /

3. _ S _D_S_ S Q ? _ QD_SF T A ? _ F T A Q A ? _ AD_S 5.

4a. ? _ S ? _ ? _ S _ S _ SF A _ Aċ Bċ↓ A _ċ P A 3.

4b. _ S _ S _ S _ S _ S F A ~. B A C 3.

E_2

1. S A↓✳A B ? AD_SA C AD_SA B 3.

2. S Q AD_SP A ?D_S A B 4 /

3. Y PF AD_SU A 5.

4a. S P AD_SA C 5.

4b. S PF A↓✳A↓✳ B A↓✳ 2 /

E_3

1. _ S _ S _ S _ S ? A ↓ ✳ A 4 /

2. _ S _ S _ S ? _ Q A I A D_S A B ċ C D_S 4.

3. _ S _ S Q A S A E A C Q S A 5.

4a. _ S _ S _ S F P A X C 5.

4b. _ S _ S _ S _ S _ ? _ Q P F A ↓ ✳ P A ↓ L ✳ P A 2 /

E_4

1. _ S _ S _ S _ S X ċ S A C A ↓ ✳ E A 2 /

2. _ S _ S _ S A ? ↓ ✳ ↓ Q ↓ X ✳ ↓ ✳ S A U D_S A ↓ ✳ A ? I A ↓ ✳ ↓ ✳ ċ 2/

3. _ S _ S I E X A D_S A 5.

4a. _ S _ S _ S _ Y F A 3.

4b. _ S _ S _ S _ S F P A ↓ D_S ✳ A ↓ ✳ A B 3.

E_5

1. _ S _ S _ S _ ? S ? ? _ S ? _ A S A↓✳ A ⌒. D_S↓ X ?
   S _ I B 2 /

2. _ Q _ Q _ S _ S ? ? _ S S X Q S _ D_S↓✳↓ X S ? _ ?
   X A↓✳A X↓✳X A 2 ? _ S /

3. _ ? _ S _ S U Q ? _ Q D_SA U A↓ A 5.

4a. _ S ? _ F P A ? _ ? S ? P A ? Q A _ C 3.

4b. _ S _ S _ S ? F X A ⌒. A Q ? A ? 2 5.

E_6

1. _ S _ S _ S _ S P A _ P U A C ⌒. A 2 /

2. _ A _ A _ A _ Q _ S _ S _ S P U A↓✳ A T A L A↓✳ A
   ↓¿ _ S↓✳T↓✳T A B T X↓T A T A B↓ D_S B 2 /

3. _ S _ S A B A U A Y A 1¿ P A T A B D_S B _ Z _ 2 /

4a. _ S _ S _ S _ S _ F P A D_S Z Q C 3.

4b. _ S _ S _ S _ S F T A B T A Q B A X A _ C _ 2 /

E_7

1. _ S _ S _ S _ S A X A _ I 3.

2. _ ? _ A _ S _ S _ S _ S P ? _ A ↓ ✳ A X ? _ A ↓ ✳ Y X ✳ A ↓ 5 /

3. _ S _ U _ S _ P D_S P_L Q ? B 5 /

4a. _ ? _ S _ S _ S _ S _ P A C Q F C B 3.

4b. _ S _ S _ S _ S ? F P A ↓ _ ¿ B F A ¿ C _ 5 /

E_8

1. _ S _ S _ S _ S ? _ P A B D_S A _ C B A D_S 4.

2. R _ A _ S _ S D_S _ S A ? _ Q A ? X _ Q ↓ ✳ A ? _ ↓ ✳ D_S ↓ ✳ X _ X ↓ ? _ B 2.

3. _ S _ U D_S A _ S P F A ↓ ? B _ Q D_S E U A Q Y D_S E 5.

4a. R P _ Q R _ S _ S ↓ ✳ F _ S _ S U ~. A U ↓ ✳ P A ~. B 5.

4b. R _ Q _ S _ Q S _ S _ S ? _ F P A ? _ Q B 5.

E_9

1. _ S _ ↓ ✳ S _ S _ S PF _ ↓ ✳ A ? Y ? _ A ↓ ✳ A ↓ ✳ A ↓ ✳ A ¿ C _ ? _ A ↓ ✳ A ¿ B ¿ B ↓ ✳ A ¿ 5 /

2. _ S _ S _ S _ S _ A ? _ A ↓ ✳ A ↓ ✳ A _ A _ C S _ ¿ S A ? _ ¿ I B ↓ ✳ A _ 5 /

3. _ ? _ S _ S _ P _ A _ 5 /

4a. _ S _ S _ S _ P ¿ ? _ P _ ? R _ 5 /

4b. _ S _ S _ S _ S _ F P A _ 5 /

E_{10}

1. _ S _ S _ S _ S ? _ A U A ↓ ✳ A E A ↓ ✳ A 2.

2. _ S _ S _ S _ S A ↓ ✳ D_S A ↓ ✳ A ↓ ✳ A ↓ ✳ D_S A B 4.

3. _ S _ S _ U D_S A E A D_S A S Q B 3.

4a. _ S _ S _ S _ PF A ↓ ✳ A B 3.

4b. _ S _ S _ S _ ? _ S PF A X ↓ _ B 5.

E_{11}

1. _ ċ R S _ S _ ? _ A ċ ? P A ċ A _ A _ A _ A ↓ ✳ A _
   ⌒. ċ B A ? _ A ? _ A ✳ A _ A _ ? 5.

2. _ S _ R _ S _ ? _ I P _ P A ? _ ? I _ _ 2/

3. _ S _ S R _ A P A ⌒. _ ? _ 2/

4a. _ S _ S _ S F _ P _ A _ _ ċ T _ A ċ C _ 3.

4b. _ S _ S _ F Q P A _ ? _ ? _ A ? _ A ċ 5 /

E_{12}

1. _ S _ S _ S _ S _ ? P A ? A ↓ ✳ A ↓ ✳ A 1 /

2. _ _ S _ S _ S _ ? A ↓ ✳ P A ? U A ↓ ✳ A ? _ A ↓ ✳ ?
   S A ↓ ✳ A ċ ↓ ✳ A _ A ↓ ✳ A ↓ ✳ ċ S _ A _ ↓ ✳ A 1 /

3. _ S _ S _ ? _ A ⌒. A ↓ ✳ A _ ↓ ✳ _ ċ U A _ 5 /

4a. _ S _ S _ ? _ S _ S _ F ? _ A ⌒. B ? _ ? _ ? P A _ 5.

4b. _ S _ S _ S _ S _ ? _ F ? ? _ A C B F A ? P A 5 /

E_{13}

1. S A ↓ ✳ A ↓ A ✳ A ↓ ✳ A ↓ ✳ A 2.

2. S ? ċ X A Q S X A Q A Q A B A B A 5 /

3. S P F Y A F A Q B A 5.

4a. S Q S F P A B 3.

4b. S F P A X A ↓ ✳ ? A ↓ A Q B 5.

E_{14}

1. S A P ⌒. A A ċ L A D_S C U T C A L D_S C Z ↓ ✳ C D_S 3.

2. S A ↓ ✳ U ? ? ↓ ✳ A ↓ ✳ I A D_S ↓ ✳ ? U D_S A D_S ↓ A D_S C B A D_S A D_S A 4.

3. S F P S F Q F Q F Q A 5.

4a. S ? P F 5.

4b. S ? P F Q P ? ⌒ · ↓ ✳ A B P A ↓ ✳ B P A Y A ↓ ✳ B ċ B A D_S A B P A C 5.

E_{15}

1. _ S _ S _ S _ S _ P A↓ ✳ A _ B Z 4.

2. _ S _ S _ S↓ ċ ↓ ✳ ċ S A↓ ✳ A ċ ↓ _ A↓ P F A↓ I↓
A↓ A↓ A↓ P↓ A↓ A↓ A↓ A↓ X I _ Q _ Q _ 3.

3. _ Q _ S _ S E A↓ B D_SA Z ⌒ 5.

4a. _ S _ S _ S _ S ? _ ? _ P F A 3.

4b. _ ? _ _ S _ S _ S _ P S F P A↓ ✳ A↓ ✳ A ? A↓ ċ Q
P A↓ ✳ P A↓ ✳ A↓ ✳ P A C 3.

C_1

1. S A↓ ✳ A B A P A B Z B↓ ✳ B↓ ✳ ċ 4.

2. S↓ ✳ Q A ? ? A I D_SA D_SA S D_S Z U I ċ A B P B I B C ċ
2 /

3. S Y P D_S A 5.

4a. S P A ? ċ ? P A 4.

4b. P F S ? ? Q S P A ? Q C 5.

C_2

1. _ S _ S _ S _ S _ A ? _ A ċ A ↓ ↓ * Q _ 1/

2. _ S _ S _ S _ ↓ ċ S * A ↓ * _ A ? _ A ↓ * ċ U A ↓ ↓ * D_S A _ 1/

3. _ S _ S _ S _ P A ? ċ _ 5/

4a. _ S _ S _ S _ S _ F ? _ P A ċ B P A _ S C _ 3.

4b. _ S _ S _ S _ S _ F ? _ ? P A B P A _ 5/

C_3

1. _ S _ S _ S _ Q S X ↓ * ↓ * A 2/

2. _ S _ ↓ * S _ Q _ Q _ S A X ⌒ . ? ↓ A 2/

3. _ S Q _ S S D_S X A 5.

4a. P F ? A 5.

4b. P F A X A 5.

C_4

1. _ S _ S _ S _ S _ A ? ↓ ✳ Y A B _ A _ 2/

2. _ ? _ S ↓ ✳ _ ? _ P A ↓ A ↓ ∩. T C ∩. ↓ ✳ ↓ ∩. A ↓ ✳ A ↓ ✳ A ↓ ✳ A ↓ ✳ A ↓ ✳ A ↓ ✳ X _ 2/

3. _ ? _ ? _ Q _ S _ S _ A Y B _ S A ∩. B _ I Q Y Z Q Z _ D_S I B _ C Y A Z C A 2/

4a. _ S _ S _ S _ D_S S ? _ F _ S _ P A ↓ ✳ A ∩. Z P A ? A _ C A C Q C A B _ ? _ I 3.

4b. _ ? S _ S _ S _ S _ S F _ A ↓ ✳ A ? P A ↓ ✳ _ Q Y P A C _ P _ Z _ ? B 2/

C_5

1. _ S _ S _ S _ S _ A ? _ A ? _ ? _ A _ ? P A _ 2/

2. _ A _ S _ S _ A ↓ ✳ A ? _ ↓ ✳ A ? _ ? _ ? _ ¿ ? _ A D_S A ↓ ✳ ↓ ✳ ? _ 1/

3. _ S _ S _ A F _ ? ¿ A ¿ Z ? _ 1/

4a. _ ? _ S _ A F A B _ 3.

4b. _ S _ S ? _ S _ F P A _ A 3.

C_6

1. _ S _ S _ S _ S ? _ A F A 4.

2. _ ? _ S _ S _ S _ ? _ S X A↓ 5/

3. _ S U _ S _ ? S _ E A D_S A∿ A 5.

4a. _ S _ S _ S _ F P U A↓ ✳ A∿ 3.

4b. _ S _ ? _ S _ ? Q S _ S F P A ↓ ✳ A X ↓ 2/

C_7

1. _ S _ S _ S A 1 A 2/

2. _ Q S _ S _ ? _ Q S _ S ↓ X ? _ ? _ 2/

3. _ S _ S A E A ? _ A D_S 5.

4a. _ S _ S _ S _ S ? F P A _ B 3.

4b. _ S _ S _ Q S _ S F P A 3.

C_8

1. _ S _ S _ S _ S ? P F A ? A ¿ Z A ¿ _ A D_S ¿ 4/

2. _ Q S _ S _ S ? _ ? _ A ↓ ✳ ↓ ✳ X A ? _ A 4/

3. _ ↓ ✳ S _ X Q _ S E Q A Y Q A ↓ D_S ? 5.

4a. _ S _ Q S _ S _ Q S ? P A 3.

4b. _ Q S _ S _ S _ S P A ? 5.

C_9

1. S Q S ¿ A P A 3.

2. S ? ? S ↓ ✳ ¿ ↓ ✳ F A ? A ? ¿ ↓ ¿ ✳ I A ? 2/

3. S ? A C Q A Y ? S Q S D_S 5.

4a. R S Q A ? ¿ P 5.

4b. Q ? ¿ S ? A ? F ? P A ¿ ? A ↓ ✳ A D_S A ? ? ? ¿ ? A ? 2/

C_{10}

1. _ S _ S _ S _ S _ ? _ P ? _ P A _ ? P ? ċ A _ P A _ A _ A _ A ċ A _ 1/

2. _ ↓ ✱ S _ S _ S P ↓ ? ċ ↓ ✱ P ? _ ↓ ċ ↓ ✱ P ? ↓ ✱ _ P _ ↓ ✱ ċ S ċ S P ? ċ _ P ċ P A ċ ↓ ċ ↓ . ↓ ✱ _ A _ ? _ A _ A ċ S _ A C ċ U ? _ ċ P I ↓ ✱ ċ A _ A ? _ A U _ ? A P ċ A ↓ ✱ A ? _ 5/

3. _ S _ S ↓ ✱ S _ S _ P A I _ P ? . _ P D_S _ D_S T _ 5.

4a. _ ? P _ S F ? _ F _ S _ S _ S _ S _ P A _ ċ C B _ 3.

4b. _ S _ S _ S _ S _ F P A C B _ 3.

C_{11}

1. _ S _ S _ S _ S _ ? _ A E A 4/

2. _ S _ S _ S _ S A ? U D_S A ↓ ✱ D_S A ↓ ✱ A B A Z B A S B I D_S B 4.

3. _ S _ S F Y A 5.

4a. _ S _ ↓ ✱ S _ S F P A _ S 3.

4b. _ S _ S _ S F P A _ S _ S 3.

C_{12}

1. _ S _ S _ S _ S _ A _ A _ C Y T A ċ B ✳ B 4.

2. _ S _ S F I A ? _ A Y A ċ S Y _ ċ ✳ A ċ D_S I A ✳ T A 4.

3. _ P _ S _ ċ D_S _ P ċ 1/

4a. _ S _ S _ S F P ? _ A _ Q C _ S _ S _ F _ P ċ Q A _ A ċ B 4.

4b. _ S _ S ċ P _ D_S _ P _ P _ ? _ ~. P D_S _ _ ċ _ P A 1 4.

C_{13}

1. _ S _ S _ ↓ ✳ S _ S _ A ? A ċ A ↓ ✳ A ↓ ✳ A C _ A ↓ X 5/

2. _ S _ S _ ? _ S _ S _ ? _ ? _ ? _ ? _ ? _ F _ A ↓ ✳ C _ 1/

3. _ S _ S _ A ↓ A _ 5/

4a. _ S _ S _ S _ S _ ? _ F _ P A _ 5/

4b. _ S _ S _ S _ S ? _ F _ P _ S _ A 3.

C_{14}

1. _ S _ S _ S _ S _ ? A _ A ċ 1/

2. _ S _ S _ S _ S _ S _ T ċ C 1/

3. _ S _ S _ A _ 1/

4a. _ S _ S _ S _ S _ F _ P A _ 5.

4b. _ S _ S _ S _ S _ F U A 2/

C_{15}

1. _ S _ S _ S _ S A ↓ ✳ _ ? _ X _ ? _ A ċ 3.

2. _ A _ S _ S _ S _ ? _ X ↓ _ S _ ✳ A X ↓ ċ ċ ✳ S _ ? A ↓ ✳ A Y A ↓ ✳ _ A ↓ ✳ A ↓ ✳ _ S _ A ↓ ✳ ↓ ✳ A ↓ ✳ A ↓ ✳ A B L C P A ċ C _ A P B _ 3.

3. _ S _ S _ S _ Q E X ċ S F A 2/

4a. _ S _ S _ S _ S P A 3.

4b. _ S _ S _ S _ S F A ↓ ✳ A 2/

APPENDIX K

Individual Subcategory Scores from Riedesel Attitude Survey

Individual Subcategory Scores from

Riedesel Attitude Survey

Experimental Group	p=4 Overall Attitude	p=48 General Attitude Toward School	p=36 Liking for Math
E1	3	39	24
E2	4	44	35
E3	4	41	32
E4	3	44	31
E5	3	46	27
E6	4	36	35
E7	4	44	33
E8	3	36	28
E9	4	30	25
E10	4	38	36
E11	3	41	29
E12	4	41	27
E13	4	48	36
E14	4	38	28
E15	3	42	29

p = possible score

Individual Subcategory Scores from

Riedesel Attitude Survey

Experimental Group	p=44 Usefulness	p=40 Anxiety	p=20 Interest
E1	40	17	13
E2	39	13	16
E3	41	17	17
E4	30	12	13
E5	41	14	14
E6	34	15	17
E7	36	22	16
E8	31	17	15
E9	34	32	12
E10	39	10	18
E11	42	18	17
E12	41	18	17
E13	44	10	20
E14	38	16	18
E15	35	17	13

Individual Subcategory Scores from

Riedesel Attitude Survey

Experimental Group	p=44 Ability & Achievement	p=8 Understanding	p=16 Preferences
E1	28	7	9
E2	35	8	10
E3	35	8	12
E4	32	7	8
E5	32	8	10
E6	27	6	13
E7	34	8	11
E8	29	7	10
E9	30	6	9
E10	39	8	12
E11	28	7	12
E12	34	8	10
E13	40	8	13
E14	34	6	13
E15	28	7	11

Individual Subcategory Scores from

Riedesel Attitude Survey

Control Group	p=4 Overall Attitude	p=48 General Attitude Toward School	p=36 Liking For Math
C1	4	25	29
C2	3	34	25
C3	2	35	18
C4	3	35	26
C5	4	31	33
C6	4	38	35
C7	2	38	22
C8	3	40	27
C9	3	43	23
C10	4	41	36
C11	4	47	36
C12	3	45	29
C13	4	38	31
C14	4	36	35
C15	4	40	26

p = possible score

Individual Subcategory Scores from

Riedesel Attitude Survey

Control Group	p=44 Usefulness	p=40 Anxiety	p=20 Interest
C1	36	33	18
C2	42	20	8
C3	32	22	9
C4	35	20	15
C5	44	10	19
C6	44	16	18
C7	35	20	14
C8	40	20	14
C9	40	19	14
C10	40	10	18
C11	41	10	19
C12	41	13	15
C13	41	17	18
C14	35	10	17
C15	35	18	20

Individual Subcategory Scores from

Riedesel Attitude Survey

Control Group	p=44 Ability & Achievement	p=8 Understanding	p=16 Preferences
C1	21	6	8
C2	32	6	7
C3	22	6	7
C4	27	4	11
C5	32	8	12
C6	27	7	11
C7	31	7	8
C8	27	6	10
C9	25	8	9
C10	33	8	13
C11	40	7	12
C12	36	8	13
C13	35	8	11
C14	31	7	11
C15	31	8	13

VITA

Name: Honi Joyce Bamberger

Permanent Address: 5412 White Mane
Columbia, Maryland 21045

Degree and Date to be conferred: Ph.D. in Education
December, 1984

Date of Birth: October 15, 1950

Place of Birth: New Rochelle, New York

Secondary Education: Milford Mill High School
Baltimore, Maryland

Collegiate Institutions Attended:	Dates	Degree
University of Maryland	1968-1974	B.S.
College Park	1974-1978	M.Ed.
	1979-1984	Ph.D.

Major: Elementary Education -- Mathematics

Minor: Special Education

Professional Experience:

1978–present	The Park School, Brooklandville, Maryland Fourth or Fifth Grade Teacher
1983–present	University of Maryland Arithmetic Clinic Consultant
1976–1979	University of Maryland Arithmetic Clinic Diagnostician/Tutor