51

TC –

0-315-22917-9

CANADIAN THESES ON MICROFICHE SERVICE – SERVICE DES THÈSES CANADIENNES SUR MICROFICHE

PERMISION TO MICROFILM – AUTORISATION DE MICROFILMER

• Please print or type – Écrire en lettres moulées ou dactylographier

AUTHOR – AUTEUR

Full Name of Author – Nom complet de l'auteur: SOMCHAI CHUCHAT

Date of Birth – Date de naissance: July 20, 1953

Canadian Citizen – Citoyen canadien: [ ] Yes Oui [✓] No Non

Country of Birth – Lieu de naissance: Thailand

Permanent Address – Résidence fixe: 383/21 Soi Raiphulnam, Amphur Muang, Chol Buri, Thailand

THESIS – THÈSE

Title of Thesis – Titre de la thèse: Computer Programming in the Senior High School Mathematics Curriculum

Degree for which thesis was presented / Grade pour lequel cette thèse fut présentée: Ph.D.

Year this degree conferred / Année d'obtention de ce grade: 1985 Fall

University – Université: University of Alberta

Name of Supervisor – Nom du directeur de thèse: Dr. S.E. Sigurdson

AUTHORIZATION – AUTORISATION

Permission is hereby granted to the NATIONAL LIBRARY OF CANADA to **microfilm** this thesis and to **lend** or **sell copies of the film.**

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

L'autorisation est, par la présente, accordée à la BIBLIOTHÈQUE NATIONALE DU CANADA de **microfilmer** cette thèse et de **prêter** ou de **vendre des exemplaires du film.**

L'auteur se réserve les autres droits de publication; ni la thèse ni de longs extraits de celle-ci ne doivent être imprimés ou autrement reproduits sans l'autorisation écrite de l'auteur.

ATTACH FORM TO THESIS – VEUILLEZ JOINDRE CE FORMULAIRE À LA THÈSE

Signature: S. Chuchat

Date: Aug. 22, 1985

THE UNIVERSITY OF ALBERTA

COMPUTER PROGRAMMING IN THE SENIOR HIGH SCHOOL MATHEMATICS CURRICULUM

BY

© SOMCHAI CHUCHAT

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE

OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF SECONDARY EDUCATION

EDMONTON, ALBERTA

FALL, 1985

THE UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR: Somchai Chuchat

TITLE OF THESIS: Computer Programming in the Senior High School Mathematics Curriculum

DEGREE: Doctor of Philosophy

YEAR THIS DEGREE GRANTED: Fall, 1985.

Permission is hereby granted to THE UNIVERSITY OF ALBERTA LIBRARY to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

S. Chuchat ....

PERMANENT ADDRESS:

Dept. of Curriculum and Instruction
Srinakharinwiroth University
Bangkhen Campus
Bangkok 10220
Thailand

Date: August 22, 1985

THE UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled "Computer Programming in the Senior High School Mathematics Curriculum", submitted by Somchai Chuchat in partial fulfilment of the requirements for the degree of Doctor of Philosophy.

....
Supervisor

....

....

....

....

....
External Examiner

Date: Aug. 22, 1985

DEDICATION

To the memory of my beloved daughter, Alisa.

ABSTRACT

The purpose of the study was to investigate the implementation of an elective unit called Integrating Programming into Mathematics (IPM) in the Math 20 program which would teach students to program, in BASIC, solutions to typical end-of-chapter exercises, and to determine the benefits that were gained by the students in this particular type of learning situation. Because the study focused on curriculum development and implementation of the IPM Elective, students' reactions to the elective and attitude to mathematics were also studied.

The curriculum development took place in the first semester. Five grade eleven (Math 20) teachers took part in the IPM program in both semesters. Semester one saw an agreed-upon course being followed by the five teachers. Evaluation in the second semester took the form of a comparison with the group of students in the regular Math 20 classes in three other senior high schools in the same school district.

The five teachers and a curriculum consultant helped the researcher to develop two manuals for utilizing in the IPM program: Student Manual for BASIC Language, and Student Exercises for Programming in BASIC - Math 20. These two manuals can be used directly in Math 20 or as a guideline to teachers intested in using this approach at other grade levels.

The analysis of responses obtained from the students indicated that the IPM students needed more help, more attention, and more explanation from their teachers. The IPM students felt that programming was an interesting part of mathematics and preferred to do mathematics with programming exercises.

Regarding the student production, the results showed that on the average, the IPM students programmed only one exercise per session. The actual number of programs written was not of the magnitude one would have thought. Very low production actually occurred.

Concerning achievement in mathematics, the results from Math 20 Test showed that at each ability level, the mean score of the control group was significantly greater ($p < 0.05$) than that of the project group. In relating the student production and the achievement findings according to the Alberta Math 20 curriculum, it is apparent that in areas where students did a lot of programming there was a significant difference in favor of the project group in one out of the three topics and no loss in the other two. In areas where students did very little programming, there were significant differences in favor of the control group in two out of the three topics.

The findings obtained from the Problem Solving Test indicate that the control group performed significantly better than that of the project group on one problem out of the five problems. Even in areas where students programmed three or four solutions, problem solving gains were not realized in the IPM Elective.

Regarding the students' attitude to mathematics, the results indicate that the mean score for each attitude scale of the project group was greater than that of the control group with significant differences on the Usefulness and Difficulty scales.

The researcher concludes that the findings of his study do not generally support the claims made by proponents of computer programming of mathematics exercises regarding the efficacy of such activity in enhancing student achievement or problem solving capabilities. However, in areas where significant amount of programming is done better results can be expected. It appears that the activity of writing computer programs to mathematics exercises in the IPM Elective is associated with the students' positive attitude to mathematics.

ACKNOWLEDGEMENTS

The writer is deeply indebted to his advisor, Dr. S. E. Sigurdson, for his patience, understanding, guidance, suggestions and assistance throughout the entire course of the study. His view of mathematics education research has been a continuing source of inspiration and his numerous hours spent in consultation allowed the ideas of the thesis to become a reality.

The writer is very grateful to Dr. A. T. Olson, Dr. W. G. Cathcart, Dr. T. E. Kieren, and Dr. L. R. Gue for their helpful comments and advice on various aspects of the study, and for their services on both the candidacy and final examining committees. The writer also wish to thank to Dr. D. B. McLeod of the Mathematical Sciences Department, San Diego State University for consenting to serve as external examiner.

The writer extends thanks to the teachers and students of Harry Ainlay Composite High School for their coopertion and assistance through almost one year of the investigation, and acknowledges with gratitude to the Department of Secondary Education, University of Alberta for graciously extending financial assistance in the form of bursaries, research and teaching assistantships.

The writer also would like to thank Shirley Sigurdson for her carefully proofreading the final draft of dissertation.

Finally, and above all, the writer expresses his

indebtedness to his wife who also a doctoral student, Alisara Sirisri, and his son, Ake, for their cooperation, encouragement, and forbearance during the whole course of graduate study.

TABLE OF CONTENTS

CHAPTER PAGE

I. BACKGROUND AND SIGNIFICANCE OF THE STUDY 1

Introduction 1

Computer and Mathematics Learning 3

Computer and Senior High School Mathematics 3

The Elective in Math 20 7

Elective as an Opportunity for Curriculum Implementation 9

Harry Ainlay Computer Project 9

Statement of the Problem 10

Purpose of the Study 10

Significance of the Study 11

Delimitations 13

Limitations 13

Outline of the Thesis 14

II. REVIEW OF THE LITERATURE 15

Theoretical Considerations 15

Curriculum Development and Implementation 15

Problem Solving 20

Attitudes 22

Computer and Mathematics Learning 24

Computer Programming and Mathematics Learning 24

Computer Programming and Problem Solving 29

Research Studies in Computer Programming and Mathematics Learning 34

Summary..46

III. PROJECT DESIGN AND RESEARCH PROCEDURES................48

Pilot Study..48

Project Design...50

Curriculum Development and Teacher Inservice...............50

Monitoring...54

Control Group..55

Process Monitoring...56

Students' Views on the IPM Elective........................56

Student Production...60

Product Evaluation...61

Achievement in Mathematics.................................61

Problem Solving..64

Attitude to Mathematics....................................67

Summary..69

IV. CURRICULUM DEVELOPMENT AND IMPLEMENTATION OF THE IPM ELECTIVE...71

Student Manual for BASIC Language..........................71

Development..71

Characteristics..72

Guidelines...76

Comments on the Development of the Five BASIC Lessons.....................................80

Student Exercises for Programming in BASIC - Math 20 - Lesson 6-15...................................82

Development..82

Characteristics..82

Guidelines....................................84

Comments on the Development of the Student Exercises Manual....................................84

Curriculum Implementation....................................85

Summary....................................87

V. PROCESS MONITORING - FINDINGS....................................89

Students' Views on the IPM Elective....................................90

Hypothesis 1....................................90

Testing Procedure and Findings....................................90

Response to Descriptive Items in the Project Group....................................92

Response to Descriptive Items in the Control Group....................................95

Comments of the Project Students on the IPM Elective....................................96

Discussion....................................97

Student Production....................................99

Discussion....................................101

General Discussion on the Process Monitoring....................................102

VI. PRODUCT EVALUATION - FINDINGS....................................104

Achievement in Mathematics....................................104

Hypothesis 2....................................104

Testing Procedure and Findings....................................104

Discussion....................................107

Hypothesis 3....................................108

Testing Procedure and Findings....................................108

Discussion....................................110

Problem Solving....................................111

Hypothesis 4....................................111
Testing Procedure and Findings..................111
Discussion......................................114
Attitude to Mathematics.........................115
Hypothesis 5....................................115
Testing Procedure and Findings..................115
Discussion......................................117
General Discussion on the Product Evaluation.......117

VII. RESULTS, INTERPRETATIONS AND IMPLICATIONS............119
Purpose and Design of the Study...................119
Summary of Results................................120
Curriculum Development and Implementation........120
Process Monitoring...............................123
Students' View on the IPM Elective.............123
Student Production.............................124
Product Evaluation...............................124
Achievement in Mathematics.....................124
Problem Solving................................125
Attitude to Mathematics........................125
Interpretations...................................126
The Meaning to the Students......................126
The Teacher Variables............................128
Curriculum Implementation........................129
Implications for Practice.........................133
Implications for Research.........................134

BIBLIOGRAPHY...136

APPENDIX I. Student Manual for BASIC Language with Answer Key....142

APPENDIX II. Student Exercises for Programming in BASIC - Math 20 with Solution Programs....178

APPENDIX III. Student Opinionnaire (A) and Student Opinionnaire (B)....203

APPENDIX IV. Math 20 Test....209

APPENDIX V. Problem Solving Test with Scoring Procedure....221

APPENDIX VI. School Subjects Attitude Scales....229

VITA....231

LIST OF TABLES

Table	Description	Page
1.	Summary of Analysis of Student Opinionnaire	91
2.	Analysis of Variance of Student Opinionnaire	91
3.	Student Opinionnaire (A): Percentages of Responses, Items 24 to 38	92
4.	Student Opinionnaire (B): Percentages of Responses, Items 24 to 26	95
5.	Summary of Analysis of Individual Topic on Math 20 Test	106
6.	Analysis of Variance of Individual Topic on Math 20 Test	106
7.	Summary of Analysis of Ability Groups on Math 20 Test	109
8.	Analysis of Variance of Ability Groups on Math 20 Test	110
9.	Summary of Analysis of Individual Problem on Problem Solving Test	112
10.	Analysis of Covariance of Individual Problem on Problem Solving Test	113
11.	Analysis of Variance of Individual Problem on Problem Solving Test	114
12.	Summary of Analysis on School Subjects Attitude Scales	116
13.	Analysis of Variance on School Subjects Attitude Scales	116

CHAPTER I

BACKGROUND AND SIGNIFICANCE OF THE STUDY

Introduction

Recent developments in the field of microprocessing have resulted in astonishing reductions in the size and cost of computing equipment. These developments permit a close association between students and computers in the classroom that was impossible just a year or so ago.

Hill (1983) has stated:

> If you are a teacher, there is a microcomputer in your future. Perhaps you are one of the rapidly growing band of educators who have access to these remarkable, self-contained units and have integrated them into your instructional program. If not, prepare to do so because even a conservative prediction is that they are destined to have a significant influence on instruction during this decade and a revolutionary impact by the last decade of this century.
>
> The mere existence of a microcomputer in a classroom does not guarantee a quantum leap toward effective instruction or positive learning. As with any technology, a microcomputer is not good or bad in or of itself. It can be misused, inappropriately used, even neglected. But its potential as an instructional tool is enormous and children surely will learn to live in a computer-pervaded world whether or not the school plays a major and guiding role in that learning. (p. 14)

Hill further states that the schools ought to become a major force in the direction of that learning. We are all challenged as teachers, administrators, and school officials to respond to the needs of children for computer literacy and computer skills, and to take full advantage of a powerful new instructional medium for concept and skill development.

There is an urgency in the challenge. Like many sectors of society, educators are being pressed by the astonishing rapidity of microcomputer development and accessibility. We need to learn from every opportunity and source so that we are able to plan for and accomplish the task of making the classroom use of microcomputers an integral, not a peripheral, part of the instructional program.

According to PRISM Canada (Priorities in School Mathematics) (Worth et al., 1981):

> Weaving problem solving experiences throughout the mathematics curriculum will develop methods of thinking and logical reasoning. To provide for this emphasis, there will need to be a reprogramming of the use of time in the classroom, more supplementary materials, greater use of calculators and computers and a shift in methodology.
>
> Every student ought to be familiar with the role and impact of the computer and how to use it. This will require that they have access to computers and calculators in elementary and secondary school classrooms and that their use be integrated into the core mathematics curriculum. (p. 61)

What are some of the objectives of school mathematics that are not being accomplished very well, and how can computers be used to help us better meet these objectives? Bell (1978) suggests that first and foremost, we must take care to set good goals and objectives for our school mathematics curriculum and that they should influence what we do with computers in schools. Conversely, he concludes that we should also be willing to extend and modify the school mathematics curriculum to exploit fully the power of computer technology in both mathematics and mathematics education.

Computer and Mathematics Learning

Papert (1980), in his thought-provoking book Mindstorms, makes an eloquent contrast. He speaks of situations in which children's contact with computers consists of the computer putting the children through their paces. This is "the computer programming the child" (p. 19). In the favored reverse relationship: "the child programs the computer" (p. 19). Instead of being instructed, "when a child learns to program, the process of learning is transformed" (p. 21), and "It becomes more active and self directed. The knowledge is acquired for a recognizable personal purpose. The child does something with it. The new knowledge is a source of power and is experienced as such from the moment it begins to form in the child's mind" (p. 21). The fundamental nature of this transformation is captured in Papert's statement: "The metaphor of computer as a mathematics speaking entity puts the learner in a qualitatively new kind of relationship to an important domain of knowledge" (p. 20).

Computers and Senior High School Mathematics

Alberta Education (1983) sets forth goals of the senior high school mathematics program in relation to three main expectations and needs: those of the individual, those of the discipline of mathematics and those of society at large. They are listed as follows:

Student Development

a) To develop in each student a positive attitude towards mathematics.

b) To develop an appreciation of the contribution of mathematics to the progress of civilization.

c) To develop the ability to utilize mathematical concepts, skills and processes.

d) To develop the powers of logical analysis and inquiry.

e) To develop an ability to communicate mathematical ideas clearly and correctly to others.

Discipline of Mathematics

a) To provide an understanding that mathematics is a language using carefully defined terms and concise symbolic representations.

b) To provide an understanding of the concepts, skills and processes of mathematics.

c) To provide an understanding of the common unifying structure in mathematics.

d) To furnish a mode of reasoning and problem solving with a capability of using mathematics and mathematical reasoning in practical situations.

Societal Needs

a) To develop a mathematical competence in students in order to function as citizens in today's society.

b) To develop an appreciation of the importance and relevance of mathematics as part of the cultural heritage that assists people to utilize relationships that influence their environment.

c) To develop an appreciation of the role of mathematics in man's total environment. (p. 18)

Looking at the goals stated above, we can see that the development of mathematical concepts, and logical analysis for an understanding of the concepts, skills and processes of mathematics subject matter should be emphasized. In school mathematics, as well as other subjects, we have not been very successful at promoting understanding of algorithmic processes, helping students learn how to learn on their own, illustrating significant applications of mathematics, teaching problem-solving heuristics, or fostering creativity. We are still unable to motivate many students to want to learn mathematics in school. However, there is much evidence in the literature to indicate that computers can and should be used in schools to help meet these important, yet hard-to-achieve, educational goals. Since good problem-solving activities foster understanding of algorithms, learning how to learn, and mathematical applications, we can summarize our consideration of these difficult-to-achieve educational objectives by asking and attempting to answer two basic questions: (1) How can using computers in classrooms help motivate students to learn mathematics? (2) How can computers be used as an aid in learning problem solving? (Bell, 1978)

A much-quoted recommendation from the National Council of Teachers of Mathematics (NCTM) Agenda for Action states: "the use of electronic tools such as calculators and computers should be integrated into the core mathematics curriculum" (p. 9). The Agenda is not defending the status quo. It warns "that just to use conventional material and techniques newly translated to the medium of the computer will not suffice" (p. 9). Drill programs may have their place but they should not be the exclusive or even major use of computers in classrooms. Programs that just transfer the demonstrations of textbook pages to the computer screen with no student interaction are a poor use of this medium. (Hill, 1983)

This means that we should rethink the curriculum to expand present goals and to formulate new goals that are now made possible by the unique character of the computer. The microcomputer expands our horizons in education. Used intelligently it can aid in reaching our objectives of helping students understand algorithmic process, and learn how to learn on their own.

Hill (1983) indicates that the most profound point to be recognized by schools and teachers is that microcomputers not only aid in accomplishing established skill and concept objectives but also they create new needs and goals for schooling. These goals relate to the particular thinking processes (organizational, systematic, and analytic) and the skills of logic and communication that enable the student to make use of technology effectively to solve problems and to live comfortably and productively in an information society and information economy.

Hill also points out that problem-solving ability and higher-order cognitive skills have always been among our educational goals. The computer provides a unique tool to aid in the solution of problems. Thus learning to communicate with and through computers, and learning to command their services in meeting human needs become essential new goals of our school programs. The limitations of classroom organization, school tradition, large classes, available media, and standardized testing, however, have made it difficult to plan systematically for their attainment. While the microcomputer is not a panacea, it certainly provides an advantage we have not had before in our attempts to develop good problem solvers.

The Elective in Math 20

One of the mechanisms to achieve the goals of the Alberta senior high school mathematics program is the elective. The elective component is a mandatory part of the senior high school program at the Grade 11 and 12 levels. The elective component primarily offers an opportunity for students to spend time on interesting and useful areas of mathematics not necessarily contained in the core and independent core components. (Alberta Education, 1983)

Alberta Education (1983) suggests the time allotment for the elective component. The elective component should be approximately 15 hours at the grade 11 level and 20 hours at the grade 12 level. It is also suggested that the 15 or 20 hours of electives be determined according to the plans of the individual teacher.

For the guidelines of the elective component, Alberta Education suggests:

1) The topics are open-ended so that the interests and abilities of students may be taken into account.

2) Student initiated projects may be considered as an elective.

3) A teacher should make use of any appropriate resources.

4) Electives should be included in the course throughout the year, wherever appropriate. These should not necessarily be taught during a 15 or 20 hour block. Where more than one elective is included, the time for each elective does not necessarily have to be the same.

5) The elective component of the program should be included in the evaluation of the students.

6) Prerequisite core material may be required before some electives are attempted. (pp. 9-10)

This elective component offers the opportunity to implement the use of computers in the senior high school mathematics program.

Alberta Education also suggests elective outlines which include computer literacy. Objectives of computer literacy for senior high school programs are stated as follows:

1. Develop an appreciation of the role of computers in our society.
2. Obtain basic understanding of the computer as a machine: its capabilities and limitations.
3. Develop the algorithmic process of thought.
4. Acquire a working knowledge of a programming language.
5. Develop skills enabling the student to use the computer for problem solving. (p. 62-63)

The objectives stated above are directly related to this particular study.

Elective as an Opportunity for Curriculum Implementation

The elective provides the researcher an opportunity for implementing a computer-based curriculum in the Math 20 program. The breadth of the interface between mathematics instruction and the computer is considerable. One particular dimension of this interface is beginning to receive some attention; computer programming of mathematics problems. The underlying premise is that using a computer language to apply mathematics concepts to problems would result in a better understanding of those concepts and development of good problem solvers. The assumption is that using only the rudiments of BASIC (Beginners All-purpose Symbolic Instruction Code), senior high school mathematics students would benefit significantly through programming the typical exercises associated with each chapter in a mathematics course. (Sigurdson, 1983)

Considering the goals of the senior high school mathematics program, and the objectives of computer literacy, this type of implementation is especially suitable to the elective being used.

Harry Ainlay Computer Project

The study reported here was carried out in conjunction with the Harry Ainlay Computer Project which was funded by

Alberta Education. The Project provided funds for the teacher inservice which made the study possible. The testing program of the present study was carried out in conjunction with the larger Project. The present study was otherwise independent from the larger Project. Further reference to the larger Project is not necessary and will not be made.

Statement of the Problem

The researcher proposed to develop and implement a curriculum of an elective unit in grade 11 mathematics (Math 20) which would teach students to program, in BASIC, solutions to typical end-of-chapter exercises; to determine students' and teachers' reactions to the unit; and to evaluate in what manner students benefit from this particular type of learning situation.

Purpose of the Study

The purpose of the study is to develop, implement, and evaluate a curriculum of an elective unit and to determine the benefits that are gained by the students. Specifically, the study searches to answer the following questions:

1. How much BASIC should be included in the Student Manual for the BASIC Language?

2. What and how many exercises should be included in the Student Exercises for Programming in BASIC - Math 20?

3. How do students react to learning mathematics in the elective curriculum? Is their reaction to their instructional setting more favorable than that of students in regular Math 20 classes?

4. How effectively do students engage in the elective curriculum as indicated by the programs that they produce and record on their disks?

5. Does the elective curriculum contribute to or detract from achievement? How well do students learn mathematics?

6. Does programming computer solutions to typical end-of-chapter exercises in grade 11 mathematics (Math 20) improve students' problem solving performance?

7. What effect does the learning of mathematics in the elective curriculum have on the attitude of students to the subject?

Significance of the Study

There are three dimensions to the significance of this study. One is the general theoretical problem investigated here of whether an experience of computer programming in mathematics improves students' performance on understanding and problem solving in mathematics. Information-processing theory and several studies suggest that this should be the case. A carefully-planned exposure for the period of a full-semester course would provide valuable data for such verification.

A second dimension of significance is the curriculum development of the study. Classroom teachers will be involved in the process of curriculum development and implementation. The study will produce two manuals which can be used by any senior high school mathematics teacher in the province:

1. Student Manual for BASIC Language - This manual will provide a 4 to 5 hour introduction to the BASIC language necessary for mathematical programming. This type of basic instruction can be incorporated into any senior high school mathematics course.

2. Student Exercises for Programming in BASIC - Math 20 - This manual will provide a collection of exercises based on the Math 20 course with typical solutions programmed in BASIC. The manual can either be used directly in Math 20 or as a guideline to teachers interested in using this approach at other grade levels.

A third area of significance at the Provincial level is to provide information for development of the "elective" component of the new Math 20 program. A challenge currently exists for teachers of senior high school mathematics to develop significant mathematical "elective" experiences for their particular students. Clearly the integrating of computers into the Math 20 course is a creative way to meet this challenge. The computer-programming component will not constitute the total "elective-portion" of the course but would be a significant contribution to it. Beyond the local level, mathematics educators world-wide are interested in the interface problem between the mathematics curriculum and the computer. Implementing the concept of programming into the senior high mathematics curriculum has not been done often and will be a real contribution to the use of computers in mathematics education.

Delimitations

The study focuses on grade 11 mathematics because the provision in the Math 20 curriculum for an elective component is the unique opportunity to test this idea. The study is also restricted to a school whose principal and teachers are agreeable to the conduct of the study.

Limitations

In the selection of the project groups and the control groups within the school, it is assumed that the previous mathematical experiences of the students in each class are similar. The cooperation and an honest effort by the students in computer programming of mathematics problems is also assumed. Another limitation is the assumption that the five classes would adequately represent all the eleventh grade students in the Edmonton Public School System.

Outline of the Thesis

The present chapter has introduced the background and the significance of the study. Chapter II provides a review of related literature and presents a rationale for the compu programming and mathematics learning. Chapter III describes the project design and research procedures. The curriculum development and implementation are explained in Chapter IV. The findings of the process monitoring and the product evaluation are reported and discussed in Chapter V and VI, respectively. Chapter VII presents a summary of the results, together with interpretations and implications for practice and research.

CHAPTER II

REVIEW OF THE LITERATURE

The present chapter is devoted to a review of literature related to the use of computer programming and mathematics learning. The review is presented in three parts. The first part discusses the theoretical basis of curriculum development and implementation, problem solving, and attitudes. The second part describes some of the literature related to computer programming and mathematics learning, and the last part describes relevant research studies that have been reported mostly in Dissertation Abstracts International.

Theoretical Considerations

Curriculum Development and Implementation

What is curriculum development? In their book, Developing a Curriculum, Nicholls and Nicholls (1972) state that teachers need to establish very clearly what they are trying to achieve with their students, then to decide how they hope to do this and finally to consider to what extent they have been successful in their attempts. In other words, the planning of learning opportunities intended to bring about certain changes in students and the assessment of the extent to which these changes have taken place is what is meant by curriculum development.

Nicholls and Nicholls' (1972) ideas of curriculum development are still true in the present education context:

> Schools and teachers are frequently criticised for the education they are providing and teachers are being encouraged, or even pressured to make changes. Most people would accept that there must be innovation of some kind. We live in a changing society in which new knowledge is constantly being discovered and in which old knowledge is being proved wrong. This problem of the tremendous increase in knowledge means that there has to be an even greater selection of what is to be learned as well as a reconsideration of how learning should take place with the realisation that students must be prepared to cope with the demands of a society which is changing so quickly. This brings with it certain responsibilities, including that of providing for students an education which is relevant to the society in which they live now and to the kind of society in which they are likely to live as adults. This means that teachers must acquire sufficient knowledge, skill and experience to make the kind of decisions which will enable them to do this. Such expertise and experience is not acquired overnight and continuing teacher education of this kind might best be achieved through actual participation in curriculum development activities, a setting in which theories are put to the test in a practical situation. (pp. 15, 17)

They also mention that the activity of curriculum development can be carried out in a variety of settings, either by an individual teacher or by groups of teachers working together. An individual teacher can undertake curriculum development for his own class in a primary school or for a subject for one or more classes in a secondary school. Obvious advantages derive, however, from either the whole staff in fairly small schools or groups of teachers in larger schools working together. There is the benefit of their joint knowledge and experience, the benefit of their complementary skills and expertise, the benefit of the ideas that are developed through interaction with each other as well as the benefit of a wider application of the results of their work than would be the case with an individual teacher, together with the possibility of expert guidance in the process of curriculum development. (Nicholls and Nicholls, 1972)

This representative field of literature in curriculum development suggests that real and effective curriculum development must go on within individual schools rather than by the creation of projects or other innovations hatched out in some central place detached from the realities of any actual school situation.

What is implementation? Fullan (1981) has described implementation as changing practice (with the emphasis on actual rather than assumed change). He expands the term more fully in saying that implementation is the process of altering existing practice in order to achieve more effectively certain desired learning outcomes for students. The term change, innovation, and revision are all frequently used in the context of describing implementation. "Change" is the generic term, with "innovation" usually referring to a more radical or thorough change than "revision." In either case, implementation is involved when a person or group of people (teachers, principals, school board administrators, etc.) attempt to use a new or revised program for the first time.

Fullan also suggests three kinds of changes as criteria for observing "changing practice" in the classroom. They are possible use of new or revised materials (direct instructional resources such as curriculum materials or technologies), possible use of new teaching approaches (i.e., new teaching strategies or activities), and the possible alteration of beliefs (e.g., pedagogical assumptions and theories underlying particular new policies or programs). All three aspects of change are hypothetically directed at achieving more effectively some new or existing educational goal.

The school culture in any country is highly resistant to change and especially to change introduced from outside the school. When 'outsiders' are able to institute curriculum changes without the cooperation and collaboration of teachers, such changes seldom go smoothly or last long. (Howson et al., 1982) The degree to which any change that we attempt to introduce into a school is likely to be effective will largely be determined by the extent to which individual teachers become committed to it. Kelly (1982) points out that there is simply no point in a school project team, a headteacher or even an enthusiastic group of teachers attempting to introduce some new scheme into a school's program unless it has the support at least of all those teachers who will need to be involved in the implementation of it and preferably a good many other teachers as well, since saboteurs can work from without as well as from within.

In particular, of course, it is vital that a project has the support of the heads of relevant departments, heads of sections within the school, other senior staff and so on.

It is not only necessary for teachers to be committed to particular forms of curriculum change, they also need to understand the purposes and the basic principles if they are to make them work properly. If any educational innovation is to be successful, the teachers must understand as well as believe in it. (Kelly, 1982)

The implications this literature has on implementation are several. First, mathematics teachers in a senior high school should participate in the process of curriculum development, together with the possibility of curriculum consultant guidance in the process. Second, meetings between the teachers and the curriculum consultant should be arranged for the teachers to learn, discuss, and understand the philosophical assumptions of the study. Other meetings should also be arranged during the developmental periods to increase teacher-teacher interaction and to share classroom experiences among the teachers. Third, the teachers and the curriculum consultant should work together in developing and implementing materials for utilizing in the actual classroom situation. Fourth, a new teaching approach for teaching a computer language and other computer knowledge should be introduced to the teachers during the teacher inservice prior to actual implementation.

Problem Solving

In their book, The Psychology of Mathematics for Instruction, Resnick and Ford (1981) have considered the various ways in which modern information-processing psychology is attempting to deal with questions of understanding and structural knowledge in mathematical learning and thinking. Their concern is to find systematic ways of representing knowledge that also account for the human capacity to reason beyond the literally given information, to find connections and relationships among different areas of knowledge, and to apply knowledge in problem-solving situations.

Information-processing theories conceive of the mind as possessing, in addition to knowledge structures, a "repertoire of problem-solving strategies" that help to interpret problems, locate stored knowledge and procedures, and generate new relations among separately stored memory items. These strategies organize the thinking process and call upon various components of knowledge to put together a plan of action capable of solving the task at hand. To account for problem solving in mathematics, then, we need to consider both the kinds of mathematical knowledge structures students have, including the kinds of algorithmic routines they are capable of performing, and the strategies they have for accessing their knowledge, detecting relationships, and choosing among the actions available to them. (Resnick and

Ford, 1981)

Greeno (1980) has discussed the problem solving process. The process of solving problems includes an important component of language understanding in which the problem solver comprehends the given information and the question. Greeno states:

> I believe that the role of a programming language in the development of a theory is analogous to the role of a general mathematical system, such as algebra or calculus, in the kinds of mathematical models that are more familiar psychology. A programming language provides a notation for representing ideas, and it provides some rather general methods for performing derivations...Theories of language understanding and of problem solving have too many component processes to permit a reliable judgment of consistency and sufficiency for observed performance unless the theory is written out in the form of a program and run on a computer. Then the computer provides a way of keeping track of the various components and their interactions. (p. 34-35)

Greeno further states:

> We also realize that someone who has learned to solve a class of problems has acquired a set of cognitive procedures including actions that change problem situations as well as procedures for setting goals and for planning, and it should be recognized that theories of language processing and problem solving already include elementary forms of learning. When a sentence is understood, new information is stored in memory, and the fact or other information expressed by the sentence is learned. When a problem is solved, new information is generated by the system and stored in memory about the specific characteristics of the problem situation and the specific goals and actions included in the solution of that problem. The learning that occurs in understanding sentences or stories or in solving specific problems is a form of assimilation, where new specific information is acquired by fitting it to existing general cognitive structures. (p. 36-37)

The BASIC language represents a limited repertoire of problem-solving strategies which is closely related to mathematical content in this particular study. The BASIC language provides notations for representing ideas, and it provides some rather general methods for performing problem solving.

In summary, from the many different perspectives through which problem-solving can be viewed, the information processing perspective seems especially useful in connecting the learning of problem solving and computers.

Attitudes

An attitude is usually an internal or a subconscious feeling about something. Attitude is not physically tangible; rather, attitude is simply a word designed to reflect how a person regards something, or how he will act under certain conditions. It is usually expressed as some set of observable behaviors. (Krulik and Weise, 1975)

Attitude has been defined in a variety of ways in the psychological literature. In this study, attitude is used to mean feelings about mathematics and feelings about oneself as a learner of mathematics. Johnson and Rising (1967) state that attitudes are fundamental to the dynamics of behavior and largely determine what students learn. In their view the mathematics student with positive attitudes studies mathematics "because he enjoys it, gets satisfaction from knowing it and finds mathematical competency its own reward" (p. 128). Johnson (1957) emphasizes this view when he says that the student with proper attitudes will enter wholeheartedly into the learning activities because he is sensitive to mathematics wherever he finds it and derives pleasure from his contacts with it.

Reyes (1980) mentions that a variety of factors influence students as they make decisions about how much mathematics to take in high school and postsecondary school. Research shows that attitudes or feelings about mathematics is an important factor in student decisions. Since their decisions about enrollment in mathematics courses are important ones and student attitudes toward mathematics affect these decisions, an understanding of these attitudes toward mathematics is essential. Not only do attitudes toward mathematics influence a student's willingness to enroll in more mathematics courses, but these attitudes also influence how much effort a student will put into learning mathematics after enrolling in mathematics classes. Beginning in the elementary school years, continuing into the middle school years, and increasingly in the high school years, students who have positive feelings about mathematics exert more effort, spend more time on tasks, and are more effective learners than students with poor attitudes.

In summary, attitude to mathematics is a relevant variable in mathematics education and one that has long term effects.

Computers and Mathematics Learning

There are many articles that support the use of computer programming in teaching and learning of mathematics. In this review, journal articles will be categorized into two categories. First, articles that relate to the effectiveness of computer programming in teaching and learning mathematics, and second, articles that relate to the use of computer programming to improve student's performances in problem-solving tasks.

Computer Programming and Mathematics Learning

In an influential article that has often been reprinted since its original publication in 1972, "Should the Computer Teach the Student, or Vice-Versa?", Luehrmann makes a strong argument that educational use of computers should primarily concentrate on students learning to program rather than on computer aided instruction. Leuhrmann argues that learning computer programming provides important experiences that enhance a student's thinking abilities and, that not to provide the opportunity of learning programming to all students in our society would be to shortchange them on their education.

Kemeny (1968), developer of the language called BASIC and mathematician identified with the well-known Dartmouth time-sharing computer center, submits that students use the computer to more effectively learn those procedures taught theoretically in class. In the publication Needed Research in Mathematics Education (1966), he states: "I feel that the right attitude is to teach them the algorithms in principle and then the right way to do the algorithm in practice is to program it for a computer. Thus a computer is being used in such a way as to force the student to explain the given algorithm to a computer. If a student succeeds in this, he will have a depth of understanding of the problem which will be much greater than anything he has previously experienced" (p. 10).

Camp and Marchionini (1984) indicate that there is a role for programming in mathematics education. As we consider the options, we must make a clear distinction between the study of programming, which belongs to the domain of computer literacy and computer science, and the use of programming to achieve learning objectives in mathematics. Programming in mathematics education is defensible to the extent that it helps achieve goals for school mathematics. But if programming is to be a general strategy for mathematics education, then particular attention must be given to format, formality, and methods by which programming is integrated into curriculum and classroom practice. The problem with programming in currently available languages is that it can be a difficult, multistep, time-consuming task. Teachers prefer devoting their time and their students' time to learning mathematics. They recognize that programming in its purest sense can potentially interfere with mathematics learning because it (1) takes too long, and (2) focuses attention on programming rather than on mathematics.

Gawronski (1981) points out that the mathematics curriculum will gradually change as a result of computer use and programming. Some content will be eliminated or receive decreased emphasis while computer programming and applications receive increased emphasis and attention. These new emphases require early introduction of the ideas of procedures and algorithms. An appreciation and understanding of the algorithmic methods is necessary to build computer-programming skills. The objective is to be able to write computer programs at different levels of sophistication, depending on the interest and talent of the individual student.

Lipson (1980) says that by writing computer programs that permit the computer to process certain kinds of information, the student in effect teaches the computer. Such programming requires precision of thought and understanding of the information to be handled - abilities that should be valuable in themselves.

Hatfield (1983) state at student programming focuses upon the learner constructing his own computer program. Extremely powerful learning experiences can occur within this context.

Tobin (1983) says:

> Students must also learn to write simple programs that are meaningful to them. Not only does this teach them that they can control a computer and tell it what to do, but they also begin to realize that

> the computer cannot do a desired task unless specific commands are given to it. Opportunities for relevant problem solving are endless. (p. 60)

Billings (1983) suggests that in writing procedures or computer programs, children actually develop the algorithm or the set of instructions needed to get the computer to do a certain task. These instructions can be spoken, written, acted out, shown by pictures or diagrams, and possibly coded and run on the computer. Children in this mode develop a sense of control over the computer and learn a variety of tools and techniques for exercising that control.

Shumway (1983) confirms:

> It seems patently obvious-and several master's studies have confirmed it - that students who write programs to do mathematics, learn mathematics. Because computers are basically dumb, and do exactly as told, students teaching computers mathematics (programming) learn a great deal of mathematics.
>
> One can be very tempted to purchase a fancy computer and to buy software so there will be "something for students to do with the computer." Don't do it - many programs that use color graphics and sound do mundane drill and practice. Students should write computer programs.
>
> Get yourself a computer; some short (less than ten lines), sample programs dealing with mathematics approachable by your students; a programming manual or two; and start your students writing programs. (p. 2)

Shumway also makes some concluding remarks: (1) All children should have the opportunity to write and run computer programs in BASIC (and LOGO); (2) Provide small, inexpensive, portable computers that can be taken home to facilitate developing computer literacy among teachers and allow as much student access to programming as is possible; (3) Computers offer children one of the few opportunities to give written instructions and see the consequences of their execution; (4) Most commercially prepared materials for computers in education will not be well received by thoughtful teachers. Simulating probability, games, science, or business is effective. Simulating thoughtful teachers is not yet possible. Do not make extensive investment in software or computer configurations that use large memory to implement software; (5) Teaching mathematics should involve 50 percent theory and 50 percent computing (computers); (6) Student programming of computers will allow for (a) computer literacy; (b) creativity in mathematics; (c) exploration of significant mathematics; (d) positive attitudes towards mathematics; and (e) marketable, basic skills for society today; (7) Do not delay, the kids are ready.

In summary, the computer is a boon to both the deductive and inductive learning process. In mathematics classes the applications and uses of the computer are many. Complex calculations are carried out, and results displayed almost instantaneously. After the computer program is developed and usable, both the teacher and students can concentrate on the mathematical structure and the physical significance of the results obtained. From the utilization of problem analysis and exploration, the student develops a feeling for what is likely to happen and the intrinsic mathematical and physical properties of the system represented by the problem. Students then participate directly in the investigation of the mathematics concepts and can learn by inference.

Computer Programming and Problem-Solving

Smith (1984) points out that instruction in computer programming can address the issues of basic skills proficiency, the lack of freshness in the middle school curriculum, computer literacy, and time. Being able to write computer programs is a significant component of computer literacy; thus programming as an integral component of the mathematics instruction necessarily contributes to computer literacy. In the process of writing programs, students explore and develop their own algorithms for solving all kinds of problems. This is an extension of their basic skills. They must also work a number of computational exercises to make sure the computer is giving the answer they want - thus gaining practice. Many facets of the basic skills of problem solving are either taught or practiced in computer programming. Trial and error, an important problem-solving technique, is seldom used by students with conventional textbook problems, but it is very effective in the computer solution of certain problems. Thinking ahead is another problem-solving competency used in computer programming. Students must ask themselves at each step, "What will happen if I do this?" Once a program is written, they can be encouraged to ask, "What will happen if I change this?" The later question is the "looking back" step in problem solving. If programming is integrated with the mathematics curriculum, then the extra time spent learning to program is minimal.

Sadowski (1983) indicates that computer programming provides a unique opportunity to teach students how to use the computer as a general purpose tool for problem solving. Higher-level cognitive skills of analysis, synthesis, and translation are called upon in the process of developing algorithms for the computer. Teaching students how to analyze a problem so that instructions can be written for the computer requires teaching techniques. Knowledge about mathematics, a language for communicating with the microcomputer, and an understanding of how the computer interprets the program are necessary. An ability to reflect on one's own problem-solving strategy along with logical-thinking skill is more important than knowing a lot of mathematics or a high-level computer language. Knowing how to use this process of logically dissecting a problem-solving strategy, developing an algorithm, flowcharting, programming, and debugging is the key to an endless adventure with computers and learning.

McIsaac (1979) points that in relating computer programming and problem solving an excellent case can be made for the use of computer programming as a vehicle for the systematic algorithmic expression of the solution of certain types of problems. The development and refinement of such solutions is a form of training for analytic thinking which is applicable to broad classes of problems. Even if one doesn't subscribe to this second statement, one can still appreciate the value of computer programming as training for dealing with abstractions and abstract reasoning.

Saunders (1978) points out that writing programs to solve problems has become the principle use of the computer. More and more people, apparently, are beginning to recognize that student control of the computer to solve problems is one of the most interesting and effective modes of using the computer in education. The students seem to enjoy this problem-solving mode of computer use so much that the computer-assisted instruction movement is not missed. In addition, students and teachers have the opportunity to practice good thinking habits such as (1)step-by-step planning, (2)anticipating errors, and (3)solution checking. Saunders states further that as a teacher of mathematics, he feels that the computer provides a powerful tool for helping kids learn about mathematics. When he designs lessons he considers possible ways for using the computer to enhance students' understanding of mathematical topics. The computer activity may be creating a program, or it may be a simple execution and interaction with a stored program. Not all such activities are successful ones, but to him, using the computer adds so much more excitement and interest to the task of learning and teaching than the usual textbook and chalkboard.

Norris (1981) states:

> I would classify mathematical thinking as a form of problem solving. It involves the analysis of given information and the synthesis of the information to discovery of new facts. This type of reasoning is very difficult. I know of no better way of introducing it to students than through the use of computers. In addition, computer programs must be very logically constructed. They are, in fact, a proof of the computability of some result. It goes without saying that computers demand discipline. One cannot deviate one iota from the prescribed rules of syntax (the computer language) or else one is presented with an error message. (p. 24),

and

> My point is this: 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. In addition, it is immensely practical to learn how to program. In the next ten years, it is likely that most college students will learn how to program computers and will use the knowledge gained (the problem-solving techniques) in their jobs....The analysis required to test and debug a program is very valuable. One must understand the structure of a program in order to provide an adequate set of test data. Trying to decide why the machine did what it did instead of what you wanted it to do is a very worthwhile experience in analytical thinking. (p. 26)

Greenes (1981) indicates that in the area of problem solving, the computer offers potential beyond its uses as an instructional tool in two ways. First, the act of developing an algorithm to solve a particular problem, programming it, and refining it is itself problem solving. Second, programs can be used to simulate mathematical or physical processes to provide students with the opportunity to analyze situations normally not present or accessible in the classroom or school environment. This potential enables students to empirically develop possible solutions to problem.

Gawronski (1981) states that a computer literate individual should know how to write an original program. The computer is a problem-solving tool, and writing computer programs utilize problem-solving abilities. Programming provides a new instructional approach for teaching mathematical concepts, applying mathematics in meaningful settings, and increasing opportunities for students to develop, use, and practice their problem-solving skills.

Inman (1981) says that the teaching of problem-solving has been one of our most difficult tasks. Problem solving is more than just a single skill. The ability to solve problems quickly and efficiently calls on several elusive skills that must be combined in an organized manner. The computer is a tool that strengthens the student's skills in problem solving. In fact, it might be said, forces students to use good problem-solving techniques.

Bozeman, Altschuler, Dorazio, and Spuck (1982) indicate that the computer has an important application in the classroom in the teaching of algorithmic analysis, the step-by-step procedure necessary to solve a problem. The need for the clear definition of a problem solution using the computer makes mathematics and science a dynamic and individual experience for the student. It eliminates the tedious calculations that may be required to open new areas of investigation.

In summary, the computer is recognized today to be a major factor in education. It has been found to be an excellent tool in the teaching-learning process. In the problem-solving mode students develop their own computer programs for solution of a problem. In analyzing it for a computer solution, the student gains a deeper understanding of the problem and the algorithm for its solution. This approach generally enhances the student's attitude and motivation to learn mathematics concepts.

Research Studies in Computer Programming and Mathematics Learning

Kieren (1968) reported a two-year study conducted with eleventh grade mathematics students in the Computer Assisted Mathematics Project (CAMP) in Minnesota. The 36 students from the first year and the 45 students involved in the second year were randomly assigned to either a computer or a non-computer group. The difference in the treatments was that the computer class learned much of their mathematics by writing BASIC programs which involved the problems, concepts, and skills from the regular mathematics course while the non-computer group did not use the computer in any way.

Kieren reported one rejection out of eight null hypotheses of no treatment effects during the first year. This rejection favored the mean of the computer class on the standardized Contemporary Mathematics Test, Advanced Level. No significant differences of treatment by previous achievement level interaction were found for the eight tests.

During the second year, the null hypothesis of no differences due to treatments for the means on trigonometry tests was rejected in favor of the non-computer class. As in the first year, the test of interaction revealed no significant differences. However, an inspection of the cell means suggest that the computer seemed to be relatively more effective for students of average previous achievement.

Hatfield (1969) completed a second study as a part of the CAMP project. Hatfield's research involved seventh graders over a 2-year period. The design and instructional procedures are analogous to those described for Kieren's study. Special supplementary materials were written and used with the computer group to teach BASIC programming, to identify the content to be programmed, and to guide the writing of programs, the study of output and the processes of "debugging" or refinement toward a more general algorithm. Comparisons involved several constructed tests, commercial standardized tests, and a selected problem solving test.

During Year 1, the effect due to treatment as measured by group means was significant for only one (Numeration System) of the 11 criterion tests. This difference favored the noncomputer treatment with the greatest difference in cell means occuring at the low previous achievement level.

During Year 2, the analysis of treatment effect revealed significance on one (Elementary Number Theory) of the six unit tests and two (Contemporary Mathematics Test and Thought

Problems) of the six post-treatment tests. These significant differences all favored the computer treatment. Comparisons of cell means on these three tests revealed that the High and Average previous achievement computer groups were especially favored.

Hatfield (1969b) concludes that the results of these two studies (Kieren's and Hatfield's study) do not support computer-assisted problem solving as the optimal approach to be taken in all settings. At the same time, there is evidence that these 7th and 11th grade students could learn to program the computer to study their school mathematics.

Ronan (1971) conducted a study in Michigan. Students of two algebra classes participated; one group of 14 boys and 12 girls who used the computer terminals and the BASIC language, and a second group of 14 boys and 11 girls who did not use the computer. The nineteen-week study had very mixed results. The computer students achieved significantly higher than the noncomputer students in one topic (exponential functions and logarithms), significantly lower in one topic (trigonometric identities and formulas), and no differently in three topics (radicals in equations, complex numbers, and circular functions). Ronan found that the computer students did achieve significantly higher in tests of logic and reasoning than their noncomputer counterparts.

Hoffman (1971) conducted a study at two metropolitan area high schools in Colorado. One class in each school was taught in a traditional manner without computer application, while the other was taught with computer application. The computer application involved eight computer extended instruction units. Students wrote programs in the BASIC language and used time sharing teletype terminals located in their respective buildings. Hoffman found no evidence that computer-extended instruction significantly affected generalization skills or achievement of second-year algebra students.

Katz (1971) directed research at a high school in Philadelphia in which average second-year algebra students studied mathematics in three ways. In one experimental class, the students wrote computer programs in conjunction with the regular classroom presentations of 19 different topics. Their programs were run on the computer by computer aides and then returned to the students. In a second experimental class, the students wrote programs as in the previously described class except that they ran their own programs on the computer. Both of these classes were compared with a control group that studied the same topics under traditional classroom instructional procedures. Comparisons made among the treatments based on mid-term and final examinations on the selected topics revealed no significant differences among the three procedures. The most effective use of the computer was having students write computer programs that were run by aides. Actually running their own computer programs did not appear to enhance students' learning experiences.

Jurick (1972) explored the incorporation of many types of computer activities into an analytic geometry course. The integrated course was presented to 18 tenth-grade geometry students with a mean IQ of 134 in one private school. The course consisted of a pretest, an orientation lecture, 19 analytic geometry lectures, and a final exam. The regular class instructor presented all lectures. The computer activities included the grading of daily seven-item, multiple-choice quizzes; checking two BASIC homework programs required in each session; presenting 10 brief ten-minute CAI segments; maintaining an internally stored student data bank; and generating daily student and teacher reports including prescribed activities. Some time problems were apparent. An average of only three CAI segments was used by each student because of the scarcity of available terminal time. The requirement of developing two programs each night was too demanding. Although one student did complete 29 out of the 38 possible programs, only six students tried more than 10. Nevertheless, test results suggested that the course was moderately successful and worth consideration for future comparative evaluations. The majority of students felt they learned no more than they would have in a conventional course, although they did enjoy the course.

Milner (1973) conducted a study, which lasted approximately 15 weeks with 18 fifth grade students, at an elementary school in nearby Pittsburgh. The study was designed to investigate the effects of teaching computer programming on performance in mathematics. Programming was used, primarily, as a conceptual basis for learning mathematics and not just as an end in itself. The research indicated the value and effectiveness of computer programming using the LOGO language in studying mathematics. Milner's conclusions based on the study are as follows: 1) Computer programming is an effective learning resource in terms of both cognitive and affective considerations. 2) The particular method used in teaching computer programming is less important than the definition of suitable tasks and the preparation for them. 3) The learner-control inherent in this study facilitated the acquisition of problem-solving behaviors. 4) Some of the students whose motivation was questionable in the traditional classroom were "turned on" by computer programming.

Foster (1973) conducted a study with 68 eighth grade students at a suburban public school in Minnesota. The study was conducted to investigate problem solving performance of students with regard to non-routine problem experiences in which the computer and flow chart were used by students as aids. Foster concluded that computer programming (and to a lesser extend flow charting) tends to support the development of selected problem solving behaviors. Reading skills may play a crucial role in the proper development of selected problem solving behaviors, especially with verbal problem solving experiences.

Mandelbaum (1974) conducted a study with 92 tenth grade students at a high school in Pennsylvania. Subjects in the control group received the regular course of study in mathematics. These students had no access or exposure to the computer. Subjects in the experimental group learned to use the computer as a problem solving tool and used the computer for 15 weeks in conjunction with the regular course of study. Mandelbaum's conclusions regarding the use of the computer as a problem solving tool for low performing tenth grade students were: (1) students are not adversely affected in their ability to do problems involving computation, concepts, or applications and (2) attitudes toward mathematics are not significantly improved as a result of the treatment.

Basil (1974) conducted a study, designed to investigate the effects of writing computer programs on achievement and attitude in elementary calculus, during the spring in 1973 at a liberal arts college located in southwestern Pennsylvania. These effects were determined by a comparison of data obtained from an experimental and a control group of students completing an elementary calculus course. The experimental group studied the BASIC programming language and used interactive terminals for completing an appropriate set of calculus homework problems. The control group used a desk calculator as an aid in completing the same set of homework problems. In conclusion, this study revealed that students can complete a course in elementary calculus, while simultaneously learning a programming language and gaining the experience of writing and debugging related computer programs, with no significant effects on their achievement or attitudes associated with calculus.

Andreoli (1976) conducted a study at a comprehensive high school in Connecticut. The experimental group consisted of 65 students who enrolled during the Fall semester of 1974 in four sections of an elective course in computer programming. Sixty-eight students who were enrolled in the same Algebra II and Trigonometry classes as the students in the experimental group were randomly selected to serve as the control group. Analysis of variance computed on the mathematical reasoning ability posttest scores of related treatment groups showed: (1) students who had taken the programmed instruction course in computer programming performed significantly better at the 0.05 level than students who had not taken the course; (2) no significant differences in mean performance between students who had been exposed to different feedback procedures; (3) no significant interactions between types of feedback procedure and mathematics aptitude level, reading level, or level of grade point average in previous mathematics courses; (4) significant interaction at the 0.05 level between I.Q. level and the combination of the feedback procedures of using programs with and without the solutions for reference.

Deloatch (1978) conducted a study with the students who were members of a population of disadvantaged students who were admitted through a special services program in Indiana.

The mathematics component of this program comprised a sequence of three one-semester courses which enroll a maximum of 18 students each. The first six instructional sessions for the experimental group were devoted solely to instruction in BASIC programming. The major distinction in the treatments was that the experimental group wrote and executed computer programs involving the concepts and computational procedures from the first course, while the control group did not use the computer or computer programming in the study of the same mathematical content. A total of eight programs were written by the experimental group on the topic of sets, integers, rational numbers, and basic algebra. Analysis of the data tends to support the assertion that computer-augmented instruction has a significant positive effect on the mathematical attitudes of disadvantaged students. The results of this investigation do not support the claims generally made by proponents of computer-augmented instruction regarding the efficacy of such instruction in enhancing student achievement in general, as well as on problem-solving tasks and computational procedures. It appears, however, that although the computer treatment did not signifiaantly increase student achievement, the activity of writing programs did not interfere with overall development of mathematical abilities. That is, in view of the past performances of this population of students, both experimental and control groups attained extremely satisfactory levels of overall achievement.

Payne (1980) conducted a study at a public high school in New York. Subjects comprising the experimental group were all 27 eleventh and twelfth grade students who were simultaneously enrolled in a one semester computer mathematics course and either an eleventh year Algebra Two-Trigonometry course or a twelfth year advanced algebra course. Control subjects were randomly picked students who were only enrolled in the academic mathematics courses and who had no previous computer mathematics experience. The study produced results, although not statistically significant, which indicated that students who simultaneously enrolled in a computer mathematics course and an academic mathematics course achieved more of the desired goals of mathematics education than students who only enrolled in academic mathematics courses. The analysis that came closest to being statistically significant, favoring students simultaneously enrolled in a computer and an academic mathematics course, was the analysis done with high average students in the area of inductive reasoning.

Krull (1980) conducted a study at a public elementary school in Missouri. The purpose of this study was to engage students in writing computer programs which focused on five properties in mathematics and to measure the effect of this activity on mathematics achievement and attitude toward school. One group which served as the experimental group received training in the computer programming language, BASIC, and spent several weeks developing and writing computer programs which focused on specific mathematical properties. The second group of students were enrolled in a remedial mathematics program. The outcomes of this study indicate that with respect to attitudes, there was no statistically significant difference between the groups. However, the experimental group did exhibit a trend toward a more positive attitude while the control group moved toward a less positive feeling about school. In addition, general observations revealed that students who were involved in programming activities performed better in their daily work.

Soloway, Lochhead, and Clement (1981) review certain research conducted in Massachusetts in 1979 and 1980. In that paper they review their attemps to isolate the specific factors in programming which enhance mathematical problem solving ability. In previous studies they found that a surprising number of engineering students had difficulty with very simple algebra word problems. They have also found that significantly more students were able to solve those word problems correctly in the context of writing computer programs, than in the context of simply writing an algebra equation. They obtained similar results in comparing the reading of algebraic equations within computer programs and reading of algebraic equations by themselves. Those observations suggest that computer programming may focus students' attention on the active, procedural semantics of equations.

Soloway, Lochhead, and Clement (1981) also conducted video-taped interviews with students as they solved typical algebra word problems. Analysis of that data supports the hypothesis stated above.

Soloway, Lochhead, and Clement (1981) conclude with a speculation on the impact of their results for education. They suggest that problem solving literacy can be promoted by teaching from an integrated algebra/programming curricula. Finally, based on companion research, they go on to suggest some cognitive factors important to the choice of which programming language should be taught, and how it should be taught:

> Since we believe that one of the primary benefits of programming is that it requires students to make each step explicit, we are suspicious of the blind initial use of sophisticated constructs (Pascal language) which have many implicit operations. We speculate that in fact students do not have mental models of the primitives which compose these higher level constructs. This suggests that they should be taught iteration first using the primitives, and then graduate to the complex constructions, as was often done in teaching BASIC and FORTRAN. (p. 14)

Generally, the studies have been conducted from a quantitative perspective and most have been recorded as dissertations. The tendency has been to involve secondary and post-secondary algebra and calculus courses to determine the effects computer programming mathematics has upon achievement, problem solving ability, reasoning ability, and attitude. Most of the studies were assigned the treatments as a non-computer treatment group and a computer treatment group. The majority of research have indicated no significant difference between treatment groups. Only a notable few of the studies have used treatments in which computer programming played an integral part of a total mathematics program.

Summary

This chapter has reviewed the theoretical basis for curriculum development and implementation, problem solving, and attitudes to mathematics; computers and mathematics learning; and specific research studies in computer programming and mathematics learning. They cal possibly be summarized as follows:

1. Real and effective curriculum development and implementation must go on within the actual classroom situation with the active participation of teachers in the process.

2. BASIC language provides a notation for representing ideas, and it provides some rather general methods for performing problem-solving tasks.

3. Positive attitude to mathematics is an important factor for students to learn mathematics. Their feelings or attitudes will influence their willingness to enroll as well as to put effort into learning mathematics.

4. Learning computer programming provides an important experience that enhances students' thinking abilities. Programming requires precision of thought and understanding of the information to be handled. Programming provides a new

instructional approach for teaching mathematical concepts, applying mathematics in meaningful settings, and increasing opportunities for students to develop, use, and practice their problem-solving skills.

5. Several research studies involving computer programming of secondary and post-secondary algebra and calculus courses have been reported. Results are inconclusive. The actual use of computer programming within mathematics programs in the reported studies varies considerably. Few of the studies represent curriculum development and implementation using regular classroom teachers. In general there still remains an optimistic outlook on the integration of programming into the mathematics course.

CHAPTER III

PROJECT DESIGN AND RESEARCH PROCEDURES

The study which was part of a larger project developed a curriculum for an elective unit in the Math 20 program which would teach students to program, in BASIC, solutions to typical end-of-chapter exercises. As was stated in Chapter 1, the purpose of the study is to investigate the implementation of the elective unit called Integrating Programming into Mathematics (IPM) and to determine the benefits that were gained by the students in this particular type of learning situation. Because the study focuses on curriculum development and implementation of the elective, students' attitude and reactions to the elective were also studied.

This chapter reports on a number of topics related to project design and research procedures used in the development, implementation, and evaluation of the elective. Specifically, this chapter describes the design of the pilot study, the project setting, the curriculum development and teacher inservice, the process monitoring, the product evaluation, the control setting, the null hypotheses tested, and the statistical procedure employed in the study.

Pilot Study

In February 1983, a pilot study was conducted at the

Harry Ainlay Composite High School, in Edmonton, Alberta, Canada. The purpose of the pilot study was to explore a particular use of microcomputer programming in teaching and learning of mathematics, and to determine how much class time and non-class time students used for programming typical end-of-chapter exercises. The researcher was also interested in the teachers' and students' reaction to the proposed curriculum. The availability of microcomputers at this school meant that students were able to try (run) their programs as they wrote them. The pilot study offered computer programming using BASIC to Math 20 students, and allowed them to program solutions to typical end-of-chapter exercises and to try their programs on microcomputers. Informal reports from the teacher in the pilot study indicated that students found the treatment a challenge. The pilot study suggested that BASIC could be taught in four to five one-hour periods. It did this by limiting the amount of BASIC for the students to learn. Once the instruction was completed the students were assigned weekly problems based on their typical Math 20 exercises. The pilot study investigated the types and number of problems students could be expected to handle. The pilot study led to the following conclusions:

1. That it was reasonable to teach senior high students the amount of BASIC needed for the elective in five periods.
2. That the teacher offering the program was prone to go far beyond the minimum of BASIC.

3. That the IPM program could be offered within the context of a 15-hour elective.

Project Design

The classes which participated in this main study were Math 20 classes at Harry Ainlay Composite High School. In this design, five classes of students were assigned to the project group. Math 20 students in the project group were offered computer programming using BASIC language, and were allowed to program solutions to end-of-chapter exercises and to try their programs on microcomputers available in their school.

The curriculum development took place in the first semester. Five Math 20 teachers took part in the study in both semesters. Semester two saw an agreed-upon course being followed by the five teachers. Evaluation in the second [illegible] took the form of a comparison with the control group of Math 20 students in three other senior high schools in the same school district.

Curriculum Development and Teacher Inservice

The curriculum development plan was to develop the curriculum over the period of the first semester. The researcher gave the five teachers two documents:

1. A preliminary student manual (See the final form

of this Manual in Appendix I).

2. A preliminary set of exercises (See the final form in Appendix II).

The teachers then met 10 times with the researcher and a curriculum consultant over the course of the semester to learn more about the elective and, in learning, to contribute to the development of the manual and the set of exercises.

Initially the teachers were given the philosophy of the program. The guideline statements of philosophy included:

1. The elective was to take place within a total of 15 classroom hours.
2. The first four to five hours of class time were devoted to the students learning the necessary amount for programming in BASIC.
3. The final remaining hours required students to program solutions to exercises.
4. The exercises were to be typical end-of-chapter exercises of the Math 20 course.
5. For one hour per week of the Math 20 course, students were scheduled into the computer lab which contained 15 Apple and 15 Commodore (PET, SuperPET, CBM) microcomputers.
6. The first two or three lessons could be conducted in the classroom rather than in the computer lab. Having a microcomputer in the classroom for some demonstrations could be a help but was not essential.

7. Students should be encouraged to read the student manual in that they would eventually use it as a reference.
8. This was not a course on programming in BASIC. Only enough programming would be taught to meet our specific purposes. It was not a course on how computers function, how they use binary numeration, nor how computers contribute to modern society. The elective was not to encompass computer assisted instruction, either drill or tutorial.
9. The elective was to be offered as an integral part of the Math 20 course.

The possibilities for using the microcomputer in mathematics teaching are numerous, but the limited time available for the elective (15 hours) meant that a restricted focus was kept. The elective amounted to a limited exposure to programming in BASIC plus student programming of typical Math 20 exercises. The purpose of the elective was to

1. promote understanding of mathematical concepts.
2. promote algorithmic thinking processes and problem solving.
3. illustrate a significant application of microcomputers in solving mathematics exercises.
4. learn about computers.

During the first semester teachers following a course plan taught the elective and met throughout the semester to discuss modifications to the original outline. Teachers' suggestions based on their experiences were incorporated into the Student Manual for BASIC Language (See Appendix I) and the Student Exercises for Programming in BASIC - Math 20 (See Appendix II).

The curriculum consists of two parts:

1. Introduction to BASIC language and microcomputer operation. The evidence from the pilot study was that four to five periods administered on a once per week basis was ample time to give senior high school students sufficient knowledge of BASIC to enable them to program solutions to mathematics problems. Each teacher involved in the study had become familiar with the BASIC language and in particular the BASIC language needed for programming solutions to mathematics problems.

2. Programming of end-of-chapter exercises. This portion consisted of assigning to students suitable exercises based on the particular chapter under consideration. This aspect of the treatment development relied heavily on the teachers' suggestions.

The first semester meetings with teachers focused on developing the above aspect of the treatment. This allowed the teachers to explore and consider suitable exercises for programming from each chapter. A considerable amount of curriculum development work needed to be done by the teachers and the researcher.

By the end of the first term, the Student Manual for

BASIC Language was revised and the Student Exercises for Programming in BASIC - Math 20 was set up based on the teachers' suggestions. The develop[illegible] characteristics, guidelines, and implementation of [illegible]wo manuals are described in detail in Chapter IV.

Monitoring

Monitoring of the study was so limited because of it being a part of a larger project which focused on the mathematical understanding, BASIC knowledge, computer awareness, attitude to computer, and over all achievement. Monitoring of the study occurred in two distinct modes: process monitoring and product evaluation. The first included answering the questions: How do students react to learning mathematics in the IPM Elective? How effectively do students engage in the IPM Elective as indicated by the programs that they produced and recorded on their disks? The data for this aspect was handled as outlined under Process Monitoring. The product evaluation addressed itself to the last four questions: Does the IPM Elective contribute to or detract from achievement? How well do students learn mathematics in the IPM Elective? Does programming computer solutions to typical end-of-chapter exercises in grade 11 mathematics (Math 20) significantly improve students' problem solving performance? What effect does the learning of mathematics in the IPM Elective have on the attitude of students to the subject? The design for this monitoring is specified under Product Evaluation.

In order to bring these questions into sharper research focus, they are now restated as research hypotheses in operational form. They are all stated as null hypotheses rather than as directed hypotheses since they are all of an exploratory nature. The researcher concluded that there existed a possibility of significant difference in either direction (as indicated in the related research studies in Chapter II) and, therefore, that a two-tailed test was required. Hence operational hypotheses of the form, $H_0: M_1 = M_2$ and $H_1: M_1 \neq M_2$ were deemed appropriate. It was considered advisable to apply the 0.05 level of significance to all tests. However, wherever possible, the calculated probability was also given.

Control Group

Due to the limited availability of Math 20 classes and the limited amount of time for administering the tests, comparable Math 20 classes in three other senior high schools, in the same school district as the project group, were chosen to serve as control groups. The problem of finding equivalent control groups was enormous since each school had its own standard for allowing students into Math 20. However, because a significant difference between the project and control group mean scores based on their Math 10 final scores was not found (p = 0.954) and the number of students involved was large, the control group was judged to be adequate for purpose of the comparison. Since, there were different control groups and different combinations of tests for different classes, the control group for each test is described separately.

Process Monitoring

I. Students' Views on the IPM Elective

The study is the result of five senior high school teachers of mathematics working together for a full school year. During this time the elective was developed and implemented. The researcher was interested in knowing the students' impressions on the IPM Elective. Over the second semester while the elective was being used in the five classrooms, the researcher was interested in monitoring the students' views on the instructional setting in three relevant areas. The first looked at the teachers' activity with respect to their explanations, attention, and assistance to the students. The second concerned the students' feelings about this particular mathematics learning environment apart from the teachers' role. Finally, the third focused on students' opinions about assignments or exercises and the amount of work they had to do in this area.

Instrument - Student Opinionnaire (See Appendix III)

To determine students' views concerning the IPM Elective, the researcher developed a Student Opinionnaire which was administered to all students who participated in the study. Particularly, the opinionnaire was developed in order to answer the following questions: How do students react to learning mathematics in the IPM Elective? Is their reactions to their instructional setting more favorable than that of students in the regular Math 20 classes?

The opinionnaire contains items designed to elicit students' views on various aspects of their instructional setting. Two forms of the opinionnaire were prepared, Form A for the project group and Form B for the control group. Items 1 to 23, inclusive, on both forms were identical to permit comparison of responses by the two groups. Under discussion with a professor in mathematics education, the first 23 items of the opinionnaire on both forms were then categorized into three scales: Teaching, Learning, and Exercises. The Teaching Scale contained five items concerning the teacher's activities to promote student's learning in terms of his/her explanation, attention, assistance, and homework checking such as "I would have liked more help from the teacher", and "The teacher is good at explaining mathematics". The Learning Scale consisted of 13 items regarding the student's feeling in terms of his/her doing mathematics, having difficulties with lessons and exercises, making mistakes in mathematics, having confidence in their own abilities in mathematics, liking or disliking and use of this type of mathematics, such as being able to learn much mathematics during this term, gaining confidence in doing mathematics. The Exercise Scale consisted of five items concerned with the amount and difficulty of homework and exercises, and the repetitiveness of exercises such as the teacher assigning too much work in the class and doing too many exercises which seemed boring. Additional items on each form applied to the instructional settings separately. All of the items consisted of statements to which students responded "Strongly Agree", "Agree", "Uncertain", "Disagree", or "Strongly Disagree". The researcher also included an open-ended question, hoping thereby to glean some information which would give direction to the interpretation of some of the results of the statistical analysis.

The opinionnaire was presented to a graduate student who had been a teacher in senior high school, and to a professor in Mathematics Education in the pilot study. Minor revisions were made as a result. In January 1984, the opinionnaire was administered to three classes of Math 20 students who were not participating in the study. Test Scorer and Analysis Program (Test 18) was used to estimate the reliability (Cronbach alpha) of the three scales (Mulaik, 1972). The reliability of the Teaching, Learning, and Exercise Scales were 0.73, 0.84, and 0.70, respectively. These coefficients were deemed sufficiently high to warrant confidence in the test's internal consistency. Further, Guilford (1965) mentions that it is not possible for a test to have high internal consistency and at the same time low retest reliability, except after very long time intervals for test-retest. High internal-consistency reliability is in itself assurance that we are dealing with a homogeneous test. The items on each scale are:

Teaching: 5, 7, 10, 17, 18

Learning: 1, 2, 4, 8, 9, 13, 14, 16, 19, 20, 21, 22, 23

Exercise: 3, 6, 11, 12, 15.

Hypothesis 1

There is no significant differences between the project and control group mean scores on Teaching, Learning, and Exercise Scales as measured by the Student Opinionnaire.

Control Setting

Four classes of grade 11 students in another senior high school in the same school district as the project group were assigned to the control group.

Analysis

To compare the control to the project group, one-way analysis of variance (with unequal sample sizes) (F

1976) was applied to data from these groups on the three scales. A descriptive analysis was used to analyze the additional items on both forms.

II. Student Production

Because the study developed a curriculum for an elective unit in the Math 20 course which would teach students to program, in BASIC, solutions to typical end-of-chapter exercises, the researcher was interested in knowing how effectively students engaged in the treatment as indicated by the programs that they produced and recorded on their disks. This also provided information as to how the curriculum should be improved for future development. In attempting to search for this information, 30 students' disks from five classes of the project group were randomly selected and the students' programs were analyzed according to the following categories

A. Problems

1. How many exercises were students able to handle per term?
2. What was the range of students' work?
3. What type of problems did students do? (i.e., routine or nonroutine problems).

B. Programming

4. What types of programs did students write? (i.e., generating numbers, applying formula, or more complex programs).

C. Student Variables

5. Did the amount of students' production vary with their ability levels according to their

Grade 10 mathematics final scores?

Analysis

Descriptive analysis was used to analyze the students' production.

Product Evaluation

The product evaluation will be presented under three headings: achievement in mathematics, problem solving, and attitude to mathematics.

I. Achievement in Mathematics

The learning of subject matter is a natural objective of mathematics instruction. Does the IPM Elective contribute to or detract from achievement? How well do students learn mathematics in the IPM Elective? Since the nature of some topics in the Math 20 curriculum such as Systems of Equations are more suitable for programming than others, the researcher was interested in identifying those topics where the elective had the greatest affect on student achievement. To measure students' achievement in mathematics, the typical year-end achievement test in Math 20 was used.

Instrument - Math 20 Test (See Appendix IV)

The Math 20 Test covers all aspects of the Math 20 curriculum with a range of cognitive levels focusing on fact, algorithmic, and problem solving levels. It covers all topics in the Math 20 program except the elective. It is used by the project and the control group schools as the standard measure of success in Math 20. This test consisted of two parts; 10 long answer questions, and 30 multiple choice questions. The grading of this test was done by the teachers involved using standard scoring schemes for multiple choice and long answers. In this particular study, the researcher analyzed the achievement scores according to topics in the Alberta Math 20 curriculum, i.e., Polynomials and Radicals (Item 1, 2, 3, 4, 1(L), 2(L)) (L means long answer problems), Relations and Functions (Item 10, 11, 12, 13, 4(L)), Coordinate Geometry (Item 5, 6, 7, 8, 3(L)), Systems of Equations (Item 19, 20, 21, 7(L)), Geometry (The Circle) (Item 22, 23, 24, 25, 26, 8(L)), and Quadratic Functions (Item 14, 15, 16, 17, 18, 5(L)). These achievement categories accounted for almost all of the items on the Math 20 Test. Variations, Trigonometry, and Statistics were not included, because there were very few items in each category.

Hypothesis 2

There is no significant difference between the project and control group mean scores on each achievement category in the Alberta Math 20 curriculum as measured by the Math 20 Test.

Hypothesis 3

There is no significant difference between the project and control group mean scores at corresponding ability levels (high, average, and low) determined by their Grade 10 mathematics final scores, as measured by the Math 20 Test.

Control group

Five classes in other senior high schools, also in the same school district as the project group, were assigned to the control group. (This control group was used as the control group for the Problem Solving Test also.)

Analysis

One-way analysis of variance (with unequal sample sizes) (Ferguson, 1976) was used to test Hypothesis 2. Two-way analysis of variance (Scheffe, 1964) was performed to test Hypothesis 3.

II. Problem Solving

As reviewed in Chapter II, the literature suggests that when a student learns to communicate with the computer by writing programs to solve his/her mathematics problems, programming is not the only thing he/she will learn but that programming will encourage the development of good problem solving skills. Does programming computer solutions to typical end-of-chapter exercises in grade 11 mathematics (Math 20) significantly improve students' problem solving performance? To determine students' problem solving performance, the researcher developed a Problem Solving Test consisting of five problems which required students to give long answers to each problem.

Instrument - Problem Solving Test (See Appendix V)

The development of the Test may be described as follows. The original form of the Test consisted of 10 problems. The Test was presented to five classes (2 problems/class) of Math 20 students who were not involved in the study in January 1984. The answers were marked accordin to the scoring procedure (See Appendix V). For each problem, the students' answers were then categorized into six categories: Totally Correct (6 marks), Mostly Correct (4-5 marks), 50% Correct (3 marks), Mostly Wrong (1-2 marks), Totally Wrong (0 mark), and No Response (Left Blank). Based on the frequency of students' answers in each category, the problems to which few students responded and the problems that seemed to be too difficult for students were eliminated. Seven problems were left for the Test. In April 1984, two teachers, who were teaching Math 20 and were not participating in the study, were asked to consider those seven problems in order to choose five problems for the Problem Solving Test. The two teachers were reminded to keep several factors in mind: the definition of problem solving; the comprehensiveness of the problems; the estimated time required to write the test.

In developing the Problem Solving Test, the researcher chose problems whose requisite mathematics skills had already been studied by the students. Problems were chosen from topics that had been dealt with in the project setting. The items were selected from various units of the Math 20 curriculum. All of the items involved procedures for solution, not merely exercises to which a rule is memorized and applied, thereby making them problems. The students were given instructions to attempt the solution as far as possible.

Problems one to five on the Problem Solving Test utilized the Radicals, Polynomials, Systems of Equations, Quadratic Functions, and Trigonometry units of the Math 20 curriculum, respectively.

The researcher scored the Problem Solving Test according to the quality of answers. Each item was assigned a maximum score of six. The general and specific scoring procedure for each problem is outlined in Appendix V.

Each problem was designed in such a way as to enable most students to complete it in 10 minutes. Due to the limited amount of time for administration of the whole test (the teachers allowed only 20 minutes), the researcher divided the test into five subtests (2 problems for each subtest): Subtest 1 (Problems 1 and 2); Subtest 2 (Problems 2 and 3); Subtest 3 (Problems 3 and 4); Subtest 4 (Problems 4 and 5); and Subtest 5 (Problems 1 and 5). Because the responses of each problem of the Problem Solving Test did not come from the same group, the researcher had to analyze each problem individually, not by subtest. "It so happens that the reliability of a test is directly related to the length of the test". "In general, longer tests tend to be more reliable than shorter ones." (Downie, and Heath, 1974, p. 238; Sax, 1979, p. 216) So, reliability for each individual problem was not found. The subtests were randomly assigned to five classes in the project group and another five classes in the control group. Thirty to 60 responses were obtained for each problem. The number of responses was deemed to be adequate for the purpose of comparison.

Hypothesis 4

There are no significant differences between the project and control group mean scores on each of the five problems as measured by the Problem Solving Test.

Control group

The control group was the same group as used in measuring achievement in mathematics.

Analysis

To increase precision in measurement for each of the five problems, one-way analysis of covariance (Winer, 1962) was used to test the hypothesis. The covariate used was the set of Math 10 final scores in order to statistically adjust the mean scores on each of the five problems to account for initial differences between groups.

III. Attitude to Mathematics

As reviewed in Chapter II, the development of attitudes to mathematics is an important objective of mathematics instruction. What effect does the learning of mathematics in the IPM Elective have on the attitude of students to the subject? The School Subjects Attitude Scales was used in the attempt to answer this question.

Instrument - School Subjects Attitude Scales (See Appendix VI)

The Scales instrument was developed during 1978 and 1979 by V. R. Nyberg and S. C. T. Clarke of the University of Alberta. It consists of 24 adjective pairs. "Factor analysis revealed three scales: evaluation, usefulness, and difficulty. Eight bipolar adjectives were adopted for each scale" (p. 1). "The scales were developed as group (classroom) measures of students' attitudes toward school subjects. They were not intended for use with individuals" (p. 1). "The reliability of the scales varied from scale to scale, subject to subject, and grade to grade. Test-retest reliabilities with a one week interval vary from a low of 0.61 to a high of 0.86. The majority of test-retest reliability scores were in the 0.70 to 0.80 range" (p. 4). The reliability estimates (internal consistency) for the Evaluation, Usefulness, and Difficulty scales were reported by the author to be 0.91, 0.90, and 0.82, respectively.

Hypothesis 5

There are no significant differences between the project and control group mean scores on Evaluation, Usefulness, and Difficulty scales as measured by the School Subject Attitude Scales.

Control group

The control group was the same group as used in evaluation of student's opinion.

Analysis

One-way analysis of variance (with unequal sample sizes) (Ferguson, 1976) was used to compare the control group with the project group.

Summary

In this chapter, a number of topics related to project design and research procedures used in the development, implementation, and evaluation of the elective has been reported. They could perhaps be summarized as follows:

1. The pilot study suggested that BASIC could be taught in four to five hour periods by limiting the amount of BASIC for the students to learn.

2. The curriculum development took place in the first semester. Evaluation in the second semester took the form of a comparison with the control group of Math 20 students in three other senior high schools in the same school district.

3. Five classes of Math 20 students were assigned to the project group, and were allowed to program solutions to end-of-chapter exercises.

4. Monitoring of the study occurred in two distinct modes; process monitoring and product evaluation. Process monitoring focused on students' views on the IPM Elective, and student production. Product evaluation looked at achievement in mathematics, problem solving, and attitude to mathematics.

CHAPTER IV

CURRICULUM DEVELOPMENT AND IMPLEMENTATION OF THE IPM ELECTIVE

Five Math 20 teachers and a curriculum consultant helped the researcher develop two manuals for utilizing in the project. Preliminary drafts of the two manuals had been developed earlier. Further development took place in the first semester (September, 1983-January, 1984). Both manuals included student materials, teacher instructional guidelines, and solutions to exercises for the teachers' use. In this chapter, the development, characteristics, and guidelines of the Student Manual for BASIC Language and the Student Exercises for Programming in BASIC-Math 20 are described. Comments on both manuals as to how they were actually used in the classrooms in the first semester are also given. In semester two an informal observation was conducted by the researcher to see how the curriculum was implemented. This will be presented under the Curriculum Implementation section.

I. Student Manual for BASIC Language

Development

The original version of the manual was developed by the researcher. After a pilot study with four class sessions of

Math 10 students, from February to June 1983, revisions were made. During the first semester (September, 1983-January, 1984) of the study, teachers following a course plan, instructional guidelines, and a philosophy, taught the elective and met throughout the semester to discuss modifications to the course and to the manual. By the end of the first semester, the manual had been revised based on the teachers' suggestions and discussions of their suggestions.

Characteristics

To incorporate the proposed computer programming elective into the Math 20 curriculum, it was necessary to exclude some BASIC statements and BASIC functions such as READ, DATA, VTAB, HTAB, SQR. Initially the SQR (square root) function was included in the manual, later it was considered that students could do this just as easily and more informatively by using a more applicable BASIC operation, raising to a power. Decreasing the number of BASIC statements that were taught was necessary to provide sufficient time to teach the necessary introductory skills of writing programs in BASIC within the limited time available for the elective. Since the BASIC language for this manual was developed relative to the mathematics topics of the Math 20 curriculum, five one-hour-per-week periods were tested to be adequate time to teach students BASIC. Only a few programming concepts were needed to begin solving mathematical problems. The LET, INPUT, PRINT, GOTO, IF-THEN, and FOR-NEXT statements were sufficient.

The manual was set up in five lessons with practice questions to conclude each lesson. The outline of lessons (See Appendix I) specifies the content to be covered in each of the five lessons. These lessons were offered only once per week and the first three lessons were recommended to the teachers to be conducted in the classroom rather than in the computer lab. The first three lessons will be described first, followed by the fourth and the fifth lessons, respectively.

The first three lessons.

The teachers' experiences in offering the elective in the first semester resulted in the following suggestions for the first three lessons. These lessons contained five key ideas.

1. Components of a computer such as the keyboard, memory, disk drive. Because in this project, the students would learn how to communicate with the computer, and to tell the computer what to do in terms of their programming. It was considered necessary to briefly give the students fundamental concepts of how computers handle information.

2. Immediate and deferred modes. Detailed and complex calculations may be done quickly and easily in the immediate-execution mode. The screen of a microcomputer

provides the opportunity to check work. The immediate-execution mode provides a very convenient method for finding errors in deferred-execution programs. Students may work through a program displaying values of selected variables until they spot one that deviates from the expected value. At that point, they may even set the correct value and direct the computer to begin execution from that point with a statement such as GOTO 80.

3. BASIC operations and program statements. Students can write simple computer programs by using a few BASIC statements such as LET and INPUT. BASIC provides many different statements which allow students to assign values to variables. LET is a mean for students to assign values to variables. INPUT is another means for the students to provide data through the keyboard for a program to work on.

Since the students can get numbers into their programs by assigning values to variables with the LET statement, and by programming with the INPUT statement, the READ and DATA statements were considered to be unnecessary. The TAB, HTAB, and VTAB functions which provide cursor positioning on the screen, were also excluded. Although nicely formatted output is easier to read and interpret, this is not a goal of the programming component of this study; the goals are to teach the logic of a program and mathematical problem solving.

4. System commands. The system commands such as LIST and RUN should be presented separately, since there is a sharp distinction between the statement in a BASIC program

and system commands (system commands tell the computers to do something with a program rather than perform program instructions).

5. Analyzing and modifying a program. It is important for the students to completely understand how the BASIC statements work within the program structure, in order for the students to be able to write a computer program. The analyzing and modifying program section was included in the manual as an approach to teaching programming.

The fourth lesson.

Most problems amenable to computer solutions require iterative procedures. Iterative procedures were implemented in the computer programs by designing loops in this lesson. (A loop is a group of instructions which the computer executes repeatedly, until some terminating condition is satisfied.) In BASIC, IF-THEN and FOR-NEXT statements are used to handle the details of looping. The FOR and NEXT statements allow BASIC programmers to form loops with fewer statements than would ordinarily be required using the techniques of the IF-THEN statement. These statements are basically complex logical statements which students might have some difficulty with in the beginning stage. They should be treated separately.

The fifth lesson.

The main goal of the fifth lesson was the use of the disk and the disk drive. Various system commands were introduced such as INIT, SAVE, and CATALOG. The idea was that a disk was a way of retaining a program for future use, revising, and/or up-dating.

Guidelines

Because the success of the elective depended on students being able to program in BASIC after five one-hour lessons, and because the Math 20 course only allowed 15 hours, the teachers were reminded to adhere closely to the suggested guidelines for the lessons. The teachers were asked to conduct the first three lessons in the classroom rather than in the computer lab. Having a microcomputer in the classroom for some demonstrations could be of help, but it was not essential. Although most of this learning was logical and made grammatical sense, the students were essentially learning to communicate in a new language (BASIC) to the computer. While the learning of meaning of BASIC language was very important, a certain element of repetition was essential. This repetition was found (during the first semester) to be most effective when done orally. The repetition done by students writing programs on the computer was slow and unwieldy, and, therefore, not very effective.

Since the first exposure to the computer lab inevitably led to confusion for a class of 30 students, it was suggested that teachers spend 10-15 minutes explaining the concepts of each of lessons four and five in the classroom, before allowing students to work in the computer lab.

Examples of Introducing Computer Programming

It is difficult to make direct claims, but programming activities seem to offer a wealth of logical reasoning experiences for students (Shumway, 1984). By definition, computer programming is the process of creating a set of instructions to be executed by the computer. To program requires that students know what instructions the computer understands and how to communica[illegible] so that the program runs correctly. In other words, students must know a computer language (BASIC, in this particular case) and how to use it.

The experience of the pilot study and the developmental phase verified that programming simple concepts with a few BASIC statements such as PRINT, LET, INPUT, could be introduced within the first one-hour period. The IF-THEN and FOR-NEXT loop could also be introduced with students practicing the writing of very simple programs in the second one-hour period. These guidelines gave the teachers a good picture of how the introduction of computer programming could be done in their classrooms.

The following methods of introducing the concepts of the computer programs were introduced to the teachers as suggested ideas during the teacher inservices. After solving equations of the form Ax + B = C, with arbitary constant A, B and C, students can be taught how to generalize the solution. Such generalizations are part of the curriculum. Once a student has solved the equations for x, in terms of A, B and C, an algorithm has been developed for a computer solution to all linear equations of this form. Here we introduce students to computer programming.

1. The program in BASIC is:

```
10 LET A = 3
20 LET B = 2
30 LET C = 10
40 LET X = (C-B)/A
50 PRINT X
60 END
```

2. The teacher could discuss with the class how the computer executes the program. The teacher should then ask the students to follow the program carefully one step at a time to determine the output. Other simple examples could be used as necessary to ensure that students have some understanding of how a computer program is executed. Instead of re-typing the program every time we want to change the value of A, B and C, we can utilize the INPUT command. The INPUT command gives real power to the computer programmer by facilitating the interaction between the student and the computer.

Program (First Revision)

```
10 INPUT A
20 INPUT B
30 INPUT C
40 LET X = (C-B)/A
50 PRINT X
60 END
```

3. Instead of having the program end, we can have it start again without re-typing the RUN command. This introduces the GOTO statement.

Program (Second Revision)

```
10 INPUT A
20 INPUT B
30 INPUT C
40 LET X = (C-B)/A
50 PRINT X
60 GOTO 10
70 END
```

4. What happens if A = 0? Students should realize that we cannot divide by zero. If we did, an error would result. The IF-THEN statement could be introduced to avoid this problem.

Program (Third Revision)

```
10 INPUT A
20 INPUT B
30 INPUT C
40 IF A = 0 THEN PRINT"BAD DATA": GOTO 10
50 LET X = (C-B)/A
60 PRINT X
70 GOTO 10
80 END
```

5. The IF-THEN and FOR-NEXT statements are best learned by writing, analyzing, and comparing two small illustrative programs regarding their procedures, efficiency, and output, e.g.,

```
10 LET N = 1
20 PRINT N
30 IF N = 13 THEN 60
40 LET N = N + 1
50 GOTO 20
60 END
```

```
10 FOR N = 1 TO 13
20 PRINT N
30 NEXT N
40 END
```

The output of both programs is the same, i.e., numbers from 1 to 13.

The above examples, introducing programming as an integral part of the Math 2Ø curriculum, allowed students to develop a better understanding of programming solutions to exercises in mathematics. Programming concepts were developed within the context of Math 2Ø. Students could become knowledgeble about developing algorithms for computer solutions, which is one of the highest levels in the computer programming process.

Comments on the Development of the Five BASIC Lessons

In the first semester while the development of the manual was on-going, many variations existed among the five teachers in terms of their background in BASIC, and their approaches to the BASIC lessons. The researcher had a chance to work with the teachers and observe the classes for 1Ø weeks. This gave the researcher a valuable experience and a clear picture of what "actual use" really meant. The following are the results of the informal observation on the "actual use" of the manual.

1. Some teachers were afraid of the students who might know more about computers than they did. On the other hand,

it is probably true that the more familiarity the student had with BASIC the better the chances of his truly benefiting in a mathematical sense from the elective.

2. The teachers should know BASIC to the elective level and be prepared to learn more. The teachers should also be familiar with the operation of the computer in the lab.

3. Although the teachers were encouraged to stick closely to the proposed guidelines, teachers with a computing background tended to stress the programming component of the elective more and tended to go beyond the minimum proposed BASIC.

4. Some teachers approached the BASIC lessons by going through the program and discussing its features, while others gave the program to the students to copy and run on the computers, or even treated the lessons as if they were offering a purely computer programming course. As the time went by, the teachers found that the latter two approaches were very time-consuming, and the students had less understanding of the programming concepts than through the first approach. With the second approach the students simply copied the program and executed it. They were anxious to see the output show up on the computer screen, without considering the functions of the program.

5. Although the researcher felt there were obvious inefficiencies in doing so, most of the teachers preferred to have the students work in the computer lab from the beginning of the first BASIC lesson.

6. After discussion and the sharing of experiences among the teachers, the researcher, and the curriculum consultant which occurred once a week, the teachers became more familiar with the philosophy of the elective that allowed only 15 one-hour periods. They then tended to adhere more closely to the suggested guidelines.

II. Student Exercises for Programming in BASIC - Math 20 Lessons 6-15

Development

In addition to developing the manual in the first semester the five Math 20 teachers were also involved in collecting, testing, and refining exercises for this manual.

Characteristics

The manual is a collection of typical end-of-chapter exercises, the solutions of which could be programmed in elementary BASIC language. The exercises were arranged according to the Alberta Math 20 curriculum. The following are examples of the exercises that are included in the manual.

- If $h(x) = 2x^2 - 4x + 3$, write a program to evaluate $h(2)-h(-3)+h(1)$.
- Write a program to find the slope of $3y - 2x + 3 = 0$.

- Write a program to find the slope of the general linear function $Ax + By + C = 0$ and use it to determine the slope of any linear function in standard form ($Ax + By + C = 0$).

- Given two equations of the form $Ax + By + C = 0$ and $Dx + Ey + F = 0$. Write a program to determine if the system has a unique solution. The user is to input the values of A, B, C, D, E, and F.

- Write a program to make a table of ordered pairs of $y = 5x^2 - 3x + 2$ for values of x from -5 to 5 inclusive.

- Write a program to list suitable pairs which will enable you to sketch the graph of
  a) $y = 2x^2 + 3$
  b) $y = ax^2 + bx + c$

In general, easier exercises were given first within a section. Within each of the 11 topics of the Math 20 curriculum, three types of exercises were given:

1. Data-generating exercises such as solving specific equations (procedures) with given numerical coefficients, e.g.,

- Write a program to evaluate $g(x) = 5x^2 - 2x + 2$ at $g(3)$, $g(-2)$, $g(0)$.

- Write a program to find the slope of $3y - 2x + 3 = 0$;

2. Applying formulae exercises such as solving for a form of an equation (procedure), e.g., the standard form of the linear equation, where any numerical coefficients may be INPUT, e.g.,

- Write a program to find the slope of the general linear function $Ax + By + C = 0$ and use it to determine the slope of any linear function in standard form.

- Write a program to find the solution of an equation of the form $Ax + By + C = 0$.

3. More complex exercises such as solving for a pre-set

range of variables or numerical coefficients, using a FOR-NEXT loop, e.g.,

- Given $P(x) = x^2 + 6x + 5$, write a program to determine if the polynomial has zeros for x values in the domain $-10 \leq x \leq 10$.

- Given two equations of the form $Ax + By + C = 0$ and $Dx + Ey + F = 0$, write a program to determine if the system has a unique solution;

Guidelines

The teacher was reminded that there were usually several ways to program any solution. The most important criterion for a good program was whether it gave correct answers, in some acceptable format. The student was then encouraged to verify his/her own program by substituting values which gave known answers.

Comments on the Development of the Student Exercises Manual

The following are the results of informal observation in the first semester.

1. Mostly, the students were not assigned or seldom assigned to write programs for exercises in the topics Variation, Geometry (The Circle), Trigonometry, and Statistics.

2. The teachers should do more in encouraging the students to use the program they had just written to solve many other problems of the same type. For example

```
10 REM FIND MAX. & MIN. VALUE OF Y=A(X^2)+BX+C
20 INPUT"ENTER A VALUE FOR A";A
30 INPUT"ENTER A VALUE FOR B";B
40 INPUT"ENTER A VALUE FOR C";C
50 LET X=-B/(2*A)
60 LET Y=-(B*B-4*A*C)/(4*A)
70 IF A<0 THEN 100
80 PRINT"THE FUNCTION HAS A MIN. VALUE OF "Y" WHEN X="X
90 GOTO 110
100 PRINT"THE FUNCTION HAS A MAX. VALUE OF "Y" WHEN X="X
110 END
```

Students used the above program to find minimum or maximum values of other quadratic functions and most of them were able to observe that when the value of A is negative the function has a maximum value and vice-versa.

III. Curriculum Implementation

Even though the program was set after the first semester, several informal observations with regard to the running of the elective in the second semester should be mentioned.

1. The teachers tended to offer the elective differently, especially in such areas as teaching BASIC, assignment of exercises, amount of guidance in programming, grading of programs, and student helpers.

2. The elective was offered on a one-hour-per-week basis. This hour was usually spent in the computer lab. Some teachers found it necessary to assign computer exercises in the regular mathematics class hour previous to the computer-lab hour. Students need to be encouraged to think programs through before coming to the lab period. In

general, the students were not given the exercises beforehand. Ideally, programs should be written before the student comes to the keyboard to type them in.

3. The teachers found that the average Math 20 classes had four or five students with good computer backgrounds. They did not appear to pose a threat to any of the teachers. Some teachers made effective use of them, as student helpers both in the classroom and the computer lab.

4. Teachers of the elective also generally found four or five students in their classes who needed significantly more help than others.

5. Class frustration levels grew during the fourth and fifth lessons when students began to work seriously at entering and saving programs.

6. The teachers found that grading all of the students' programs was impossible. Because the students, themselves, could easily check whether a program works, another approach was to ask students to submit one program per topic for grading.

7. Having two or more types of computers in the computer lab added greatly to the confusion and frustration of lab periods. Teachers must be completely familiar with the operation of the computer installation in the computer lab.

8. Teachers felt that the amount of mathematics learned increased as the student became more competent in programming in BASIC. Students in the elective should be encouraged to

use their time to do more mathematics not to increase their competence as programmers in accordance with the goal of the elective.

Summary

In this chapter, the development, characteristics, and guidelines of the two manuals have been described. The comments on both manuals as to how they were actually used in the classroom in the first semester and how they were implemented in the second semester were also given. They can be summarized as follows.

1. The researcher developed the preliminary drafts of the two manuals.

2. A curriculum consultant assisted the five Math 20 teachers in further development of the two manuals in the first semester.

3. After the first semester, the two manuals were set and implemented in five classes of Math 20 students in the second semester.

4. The Student Manual for BASIC Language was set up in five lessons with practice questions to conclude each lesson. These lessons were offering only a one-hour period per week.

5. The Student Exercises for Programming in BASIC - Math 20 is a collection of typical end-of-chapter exercises, the solutions of which could be programmed in elementary BASIC language. These sets of exercises were given to the

students in the remaining 10 one-hour periods of the elective.

6. The teachers' previous knowledge and beliefs had a large impact on curriculum implementation.

CHAPTER V

PROCESS MONITORING - FINDINGS

The purpose of the monitoring was to evaluate the implementation of the elective unit and to determine the benefits that were gained by the students in this particular type of learning situation. This chapter reports the findings of the analysis of the data on process monitoring collected during the Evaluation Phase of the study. Process monitoring looks at the students' reactions to learning mathematics in the IPM Elective and how effectively students engaged in the elective as indicated by the programs that they produced and recorded on their disks. The findings are reported under headings corresponding to the research questions posed in Chapter III. In presenting the findings related to each question, the null hypothesis is presented, the testing procedure is described, and the findings of the analysis are given and discussed. This pattern of presenting the findings will be seen again in Chapter VI, the Product Evaluation.

All statistical analyses were carried out on the AMDAHL 580-5860 computer at the University of Alberta, using statistical programs developed by the Division of Educational Research Services, Faculty of Education at that institution.

I. Students' Views on the IPM Elective

Hypothesis 1

There are no significant differences between the project and control groups' mean scores on Teaching, Learning, and Exercise Scales as measured by the Student Opinionnaire.

Testing Procedure and Findings

Hypothesis 1 was tested using one-way analysis of variance (with unequal sample sizes) (Ferguson, 1976). The findings of the analysis are presented in Table 1 and 2.

The probability value (Table 2) is less than 0.05 only on the Teaching Scale. Thus Hypothesis 1 was rejected for this Scale. For this Scale, therefore, a significant difference was indicated between the two groups. Inspection of the mean scores reported in Table 1 indicates that the control students responded more favorably to their instructional settings on the Teaching Scale than the project students did to theirs. The project and control students' responses (mean scores) on the Learning and Exercise Scales are about the same (42.54, 42.88, and 15.21, 15.80, respectively).

TABLE 1

Summary of Analysis of Student Opinionnaire

Scale	Group	Number	Mean	Variance	S.D.	Homogeneity of Variance
Teaching	P	68	14.69	12.34	3.51	x^2 = 0.179;
	C	85	16.48	11.18	3.34	p = 0.672
Learning	P	68	42.54	64.97	8.06	x^2 = 0.423;
	C	85	42.88	75.60	8.69	p = 0.516
Exercise	P	68	15.21	12.29	3.50	x^2 = 0.024;
	C	85	15.80	12.73	3.57	p = 0.878

TABLE 2

Analysis of Variance of Student Opinionnaire

Scale	Source of Variation	Sum of Squares	D.F.	Mean Square	F Ratio	p
Teaching	Groups	121.22	1	121.22	10.37	0.002
	Error	1765.69	151	11.69		
Learning	Groups	4.11	1	4.11	0.06	0.810
	Error	10703.47	151	70.88		
Exercise	Groups	13.35	1	13.35	1.07	0.304
	Error	1892.67	151	12.53		

Responses to Descriptive Items in the Project Group

Only students in the project group responded to Items 24 to 38 of the Student Opinionnaire (A) which referred to mathematics and computer programming. Table 3 gives the percentages of responses by the students in the project group to the Items 24 to 38, inclusive.

TABLE 3

Student Opinionnaire (A)

Percentages of Responses, Items 24 to 38

Item*	Percentage of Responses		
	Agree	Uncertain	Disagree
24 Programming is mathematics	62	18	20
25 Programming & math differ	22	25	53
26 Don't like mistake with program	68	15	17
27 Have more explanation in class	62	16	22
28 Math in class without program	20	18	62
29 Work with BASIC Manual	31	47	22
30 Programming meaningful	31	38	31
31 Learn as much math in class	48	28	24
32 BASIC Manual too hard	21	34	45
33 Work from problem sheet	46	28	26
34 Frustrated with programming	31	22	47
35 Short review helpful	42	31	27
36 Later periods not fun	18	40	42
37 Confidence with programming	45	31	24
38 Programming helped math	47	21	32

* The complete statement of each item may be found in Appendix III (Student Opinionnaire (A)).

Inspection of Table 3 reveals that the results may be summarized as follows. First, concerning the importance of programming mathematics exercises, the majority of the project students (62%) thought that computer programming was very mathematical (Item 24) and more than one-half (53%) of these students thought that computer programming did not seem too different from mathematics (Item 25). Most of the students (62%) would not have preferred doing mathematics without programming exercises (Item 28), and nearly one-half (47%) of them thought that programming exercises in mathematics helped them to use mathematical knowledge to solve problems (Item 38) and also made them more sure of themselves when they returned to the classroom (Item 37, 45%). However, nearly the same proportion (48%) of these students thought that they would have learned just as much mathematics in the classroom without programming exercise periods (Item 31) and more than one-third (38%) of them were uncertain that programming mathematics exercises gave them a more meaningful way of doing mathematics (Item 30). So although they think programming is important they do not think it helps in learning mathematics. This appears to be a very important distinction that the students are making. Secondly, concerning the students' working with computer programming, the majority of the project students (68%) did not like to make a mistake when they did the programming exercises in mathematics (Item 26) and about one-half (42%) of these students thought that programming exercises in mathematics periods were fun (Item 36), but nearly one-third (31%) of them were frustrated when they did programming exercises in mathematics (Item 34). The programming activity seems to the students to be a mixed blessing: fun but challenging and frustrating. It might suggest that it is the kind of activity that could get out of hand unless treated carefully by teachers.

Thirdly, concerning the curriculum materials, the students found the curriculum materials appropriate. Close to one-half (42%) of the students thought that the short review of BASIC by the teacher before each lesson was useful (Item 35). Nearly the same proportion (45%) of the students thought that the BASIC Manual was not too hard to understand (Item 32). Concerning the teaching approach, most of the project students (62%) would have liked to have had more explanation in the classroom about programming before they went to the microcomputer lab (Item 27) and about one-half (46%) of them liked working from the problem sheets (Item 33). The problem sheets were used to help the students with learning and practicing programming, e.g., the first program might have one line blank for the students to fill in. The number of blank lines increased in the later programs. Although the students found the curriculum materials appropriate, the majority of them suggested that there should have been more in-class instruction. This might suggest that a more effective way of teaching programming would be to explain the features of programming more thoroughly in the classroom. After students are knowledgable enough about programming, working with the computers would be more enjoyable and less frustrating.

Responses to Descriptive Items in the Control Group

Only students in the control group responded to Items 24 to 26 of the Student Opinionnaire (B) which related to the students' reactions to the possibility of their having experience in computer programming in mathematics. Table 4 gives the percentages of responses by students in the control group to Items 24 to 26 of the Student Opinionnaire (B).

TABLE 4

Student Opinionnaire (B)

Percentage of Responses, Items 24 to 26

Item	Percentage of Responses		
	A	U	D
24. I would like to have some experience in computer programming.	63	14	23
25. I would like to know more about computer programming and the application of mathematics.	59	20	21
26. A mathematics course should involve computer programming.	52	20	28

From Table 4 it is apparent that the majority of the control group students would like to have some experience in computer programming (Item 24, 63%) and would like to know more about computer programming and the application of mathematics (Item 25, 59%). Over one-half thought that a mathematics course should involve computer programming (Item 26, 52%). So the control group then exhibited the same enthusiasm for programming in mathematics as the project group.

Comments of the Project Students on the IPM Elective

In the previous section students' reactions were reported under several themes. The following is a summary of positive and negative comments made by the project students which expressed their feelings regarding the themes. These comments were collected from an open-ended question on the Student Opinionnaire (A).

Positive Comments

The project students found computer programming "an interesting part of mathematics". They felt that the elective was "an enjoyable and educational experience" and "would like to continue it in Math 30". They also suggested that "computer programming should become a basic part of the Math 10 and Math 20 courses". They thought that learning how to program the computer was "fun", "useful", and "important to know". They felt that the elective "should be continued". There was "not enough time to complete the requested programs" and there should be "more time allowed on the computer than one hour a week". This time factor is an interesting observation which comes up again. They also gave a new theme of comment that students who had some computer background found the elective "rather easy and gave more hands on experience on how to communicate, (how to) interface with the computer".

Negative Comments

Although the project students found computer programming "interesting" and "useful", they thought that they "would have learned just as much mathematics in the classroom without programming the exercises". They thought that the teachers "seemed to have insufficient knowledge of computer programming" and "did not help them in preparing the programs". There should have been "more in-class instruction", and they would like it if the teachers "gave more individual help and attention". They thought that all of them "should work with the same type of computers". This type-of-computer factor is another interesting new theme in student comments.

Discussion

As was shown in Tables 1 and 2, the students in the control group responded more favorably to their instructional settings on the Teaching scale than the project students did to theirs. This occurred probably because for a significant portion of the students the elective was challenging and frustrating. Also many of the project group students who did computer programming in mathematics needed more help from their teachers, especially when they were writing, entering, and saving programmed solutions to exercises. But in spite of this reaction to the teaching, the project students generally felt that programming was an interesting part of mathematics and it should be more than one hour per week in order to have enough time to work on exercises.

Although the majority of the project students preferred to do mathematics with programming exercises, less than one-half of these students thought that programming exercises helped them to use mathematical knowledge to solve problems. This shows that most of the students did not see the connection between programming exercises and mathematics learning. They preferred to do mathematics with programming exercises simply because they thought it important and enjoyed it.

Although the control students did not have a chance to work with computer programming and to learn about the connection between programming and mathematics learning, the majority of them would like to have experience in computer programming and felt that a mathematics course should involve computer programming. Assuming the project group felt the same before the elective unit, one can safely say that the elective did not affect the belief negatively.

A good percentage of the students felt that they did not get enough help and individual attention from teachers. Because the computer demands exact commands, the students did not like to make mistakes when they did programming exercises, and therefore felt the need for more explanation in the classroom before going to the computer lab.

II. Student Production

Thirty students' disks from five classes of the project group were randomly selected and the students' programs were analyzed in an attempt to answer the question: How effectively do students engage in the IPM Elective as indicated by the programs that they produced and recorded on their disks? The students who worked on the Commodore microcomputers had problems with the operation of the muppet system which connected eight microcomputers to one dual disk drive. The researcher estimated that eighty percent of the students' programs were saved on their disks. Very few problems about saving the programs were found with the students who worked with the Apple microcomputers which had their own disk drive machines. The students' programs were analyzed and the findings were reported according to the following questions.

1. How many exercises were students able to handle per term?

It was found that on the average the students did 12 exercises per term.

2. What was the range of students' work?

The number of students' programs that were saved on their disks was between 3 and 29 programs. Ten percent of the students did 1-6 programs, 53% did 7-12 programs, 20% did 13-18 programs, and 17% did over 18 programs.

3. Did the amount of students' production vary with their ability level according to their Grade 10 mathematics final scores?

The average number of student programs in the low ability level was 12 programs, in the middle ability level it was 14 programs, and in the high ability level the average was 8 programs.

4. How much work did students do in each topic according to the Alberta Math 20 curriculum?

The percentage of students' programs in each Math 20 topic were: Polynomials & Radicals, 22%; Relations & Functions, 6%; Coordinate Geometry, 27%; Systems of Equations, 2%; Variations, 4%; Geometry (The Circle), 4%; Trigonometry, 5%; Statistics, 2%; and Quadratic Functions, 28%.

5. What types of programs did students write?

As mentioned in Chapter III, the students' programs were categorized into three types. An actual example of a program in each type follows.

Generating number program

```
10 REM EVALUATE G(X) = 5X^2 - 3X + 2
20 LET X = 3
30 LET Y = 5*X^2-3*X+2
40 PRINT"WHEN X = 3, G(3) = ";Y
50 END
```

Applying formula program

```
10 PRINT"THIS PROGRAM WILL FIND SLOPE OF"
20 PRINT"EQUATIONS OF THE FORM AX+BY+C=0"
30 PRINT"TYPE IN THE A,B,A VALUES"
40 INPUT A,B,C
50 LET D=-A/B
60 PRINT"THE SLOPE OF EQUATION IS ";D
70 END
```

More complex program

```
10 REM ORDERED PAIRS FOR ANY QUADRATIC FUNCTION
20 PRINT"ENTER THE VALUES FOR A,B,C"
30 INPUT A,B,C
40 PRINT"(X,Y)"
50 FOR X=-5 TO 5 STEP .5
60 LET Y=A*X^2+B*X+C
70 PRINT"("X","Y")"
80 NEXT X
90 END
```

Fourteen percent of the students' programs were generating numbers type, 69% were applying formula type, and 17% were more complex program type.

Discussion

It appears that on the average students did one exercise per session. It shows that 17% of the project students did over 18 programs, but more than one-half of the students (53%) did only 7-12 programs. A significant amount of students' production was done in the topics Polynomials, Radicals, Coordinate Geometry, and Quadratic Functions, but a small percentage of students' production actually occurred in the topics Systems of Equations, Variations, Geometry, Trigonometry, and Statistics. It also shows that the middle ability students did 14 programs (average), but the high ability students did only 8 programs (average). It does appear that the bright students did not do a lot of programming. Probably, the more experienced students programmed before the Elective began, the more programs they can write. In any case, the actual number of the programs was not of the magnitude one would have thought - very little production actually occurred. The researcher estimated that on the average the students should be able to write 40 programs. In Chapter VII reasons for the low production will be suggested.

General Discussion on the Process Monitoring

Although the project group students felt that programming was interesting and enjoyable, little production (solution programs) actually occurred. They preferred less their teachers' teaching than the control group students did theirs. The project students needed more help from the teachers when they were writing, entering, and saving their programs. This might be one factor that prevented them from producing a significant amount of work. They still thought of the mathematics course and the IPM Elective as separate courses. They did not see the connection between the programming exercises and mathematics learning. This led them to see it as being totally separated from the Math 20 course, less important in terms of grades which might explain what appears to be a small amount of effort given to programming.

CHAPTER VI

PRODUCT EVALUATION - FINDINGS

The previous chapter reported the findings of the analysis of the data on process monitoring. The present chapter reports the findings of the analysis of the data on product evaluation also collected during the Evaluation Phase of the study. The product evaluation concerns the contribution of the IPM Elective unit to the students' achievement, problem solving performance, and the attitude to the subject.

I. Achievement in Mathematics

Hypothesis 2

There is no significant difference between the project and control group on the total achievement mean scores on each topic in the Alberta Math 20 curriculum as measured by the Math 20 Test.

Testing Procedure and Findings

Hypothesis 2 was tested using one-way analysis of variance (with unequal sample sizes) (Ferguson, 1976). The findings of the analysis are presented in Tables 5 and 6.

Table 5 summarizes the data required for the F-tests on significant differences between means of individual achievement topics. Table 6 shows the probabilities of these differences. The hypothesis is rejected in favor of the project group for the Polynomials and Radicals topic, and is rejected in favor of the control group for Relations and Functions, and Geometry (The Circle). There were no significant differences between the two groups on Coordinate Geometry, Quadratic Functions, and Systems of Equations. An inspection of the cell means reveals that mean scores of the project group were greater than those of the control group on the Polynomials and Radicals, and Coordinate Geometry topics, while the mean scores of the project group were lower than those of the control group on Relations and Functions, Quadratic Functions, Systems of Equations, and Geometry (The Circle).

TABLE 5

Summary of Analysis of Individual Topic on Math 20 Test

Topic	Group	Number	Mean	Variance	S.D.	Homogeneity of Variance
Polynomials	P	133	9.38	13.92	3.73	x^2 = 0.023;
& Radicals	C	128	8.27	13.55	3.68	p = 0.882
Coordinate	P	133	5.90	8.76	2.96	x^2 = 0.097;
Geometry	C	128	5.58	8.29	2.88	p = 0.755
Relations	P	133	4.68	8.31	2.88	x^2 = 0.148;
& Functions	C	128	5.49	7.76	2.79	p = 0.700
Quadratic	P	133	9.08	12.58	3.55	x^2 = 1.270;
Functions	C	128	9.92	15.33	3.92	p = 0.261
Systems of	P	133	6.59	6.50	2.55	x^2 = 0.155;
Equations	C	128	6.84	6.97	2.64	p = 0.694
Geometry	P	133	8.54	9.22	3.03	x^2 = 0.275;
(The Circle)	C	128	9.67	10.11	3.18	p = 0.600

TABLE 6

Analysis of Variance of Individual Topic on Math 20 Test

Topic	Source of Variation	Sum of Square	Degree of Freedom	Mean Square	F	p
Polynomials	Groups	79.27	1	79.27	5.77	0.017
& Radicals	Error	3558.58	259	13.74		
Coordinate	Groups	6.53	1	6.53	0.77	0.382
Geometry	Error	2209.70	259	8.53		
Relations	Groups	42.58	1	42.58	5.30	0.022
& Functions	Error	2082.69	259	8.04		
Quadratic	Groups	45.94	1	45.94	3.30	0.071
Functions	Error	3607.26	259	13.93		
Systems of	Groups	4.06	1	4.06	0.60	0.438
Equations	Error	1742.91	259	6.73		
Geometry	Groups	83.36	1	83.36	8.36	0.004
(The Circle)	Error	2501.20	259	9.66		

Discussion

In general, the researcher is interested in whether the IPM Elective had any effect on the students' achievement. As shown in Table 6, the project group did significantly better on Polynomials and Radicals, while the control group did significantly better on Relations and Functions, and Geometry (The Circle). The IPM Elective had a significantly positive effect on the project group's achievement in one particular topic, but the control group was better in the two other topics. There should be some explanation for these occurrences.

It is interesting to relate the student production with the findings. In areas where students did a lot of work (as reported in Chapter V, 22% of the programs were in the Polynomials and Radicals topic, 27% in the Coordinate Geometry topic, and 28% in the Quadratic Functions topic), there was a significant difference in favour of the project group in one (Polynomials and Radicals topic) out of the three topics and no loss in the other two. In areas where students did very little work (6% of the programs were in the Relations and Functions, 2% in the Systems of Equations, and 4% in the Geometry (The Circle)), there were significant differences in favor of the control group in two (Relations and Functions and Geometry (the Circle) topics) out of the three topics. Therefore, it could expected that in the areas where students did a significant amount of work an increase in students' achievement (or at least no decrease) in that area would occur. Ideally, the minimum of three programming exercises per topic should be completed by the students in order to improve or maintain their own achievement. On the other hand, in the IPM Elective in areas where a significant amount of work was not done, a decrease in achievement scores may occur. The IPM Elective seems to detract from achievement especially in those areas where students do less than three programming exercises.

Hypothesis 3

There are no significant differences between the project and control group mean scores at corresponding ability levels according to their Grade 10 mathematics final scores as measured by the Math 20 Test.

Testing Procedure and Findings

The students in both project and control groups were divided into three subgroups, each on the basis of the Math 10 final scores. The Math 10 final scores were used to rank the Math 20 students with the 33rd and 67th percentiles marking the divisions into high, average, and low ability groups. The Math 20 Test scores were then classified two ways, by treatment groups (project and control) and by ability groups, and a two-way analysis of variance was performed (Scheffe, 1964).

The findings of the analysis of Hypothesis 3 are presented in Table 7 and 8. From the findings, it is apparent that for each ability group, the mean score of the control group was greater than that of the project group. Table 8 shows that the treatment effect was significant (p = Ø.Ø44) while the interaction effect was not significant (p = Ø.964). Hence the findings did not support Hypothesis 3, therefore, it was rejected. At each ability level, the mean scores of the control group was significantly greater than that of the project group.

TABLE 7

Summary of Analysis of Ability Groups on Math 2Ø Test

		High Ability	Average Ability	Low Ability	Total
Project	Obs. No.	44	45	44	133
	Mean	63.64	51.51	44.66	53.48
	Variance	169.82	137.71	13Ø.23	2Ø4.16
Control	Obs. No.	42	44	42	128
	Mean	67.38	54.43	47.45	56.39
	Variance	188.1Ø	154.67	173.Ø8	236.67

TABLE 8

Analysis of Variance of Ability Group on Math 20 Test

Source of Variation	Sum of Squares	D.F.	Mean Squares	F Ratio	p
Treatment	648.20	1	648.20	4.09	0.044
Ability Group	16730.00	2	8363.38	52.76	0.000
Interaction	11.56	2	5.78	0.04	0.964
Error	40420.00	255	158.51		
Total	57810.00	260	222.34		

Discussion

The results reported in this section do not generally support the claims made by proponents of computer programming of mathematics exercises regarding the efficacy of such activity in enhancing students' mathematics learning. It is apparent that at each ability level, the mean scores of the control group were significantly greater than that of the project group. Even if, as might be expected, the average ability group students did a lot of programming in Polynomials and Radicals, Coordinate Geometry, and Quadratic Functions, the improvement in these three areas would still not affect the overall achievement score since these three areas accounted for only 41% of the overall achievement score. The evenness of achievement differences across ability levels suggests that the one hour per week spent on the IPM Elective which meant one hour per week less in the regular mathematics course negatively affected all students.

II. Problem Solving

Hypothesis 4

There is no significant difference between the project and control gr iean scores on each problem as measured by the Problem Solving Test.

Testing Procedure and Findings

Although the subtests of the Problem Solving Test were randomly assigned to five classes in the project group and another five classes in the control group, the difference between the two groups for each problem due to their previous Math 10 final scores would still be expected. This hypothesis was tested using one-way analysis of covariance (Winer, 1962). The covariate used was the Math 10 final score in order to statistically adjust the mean scores on each of the five problems to account for initial differences between groups.

The findings of the analysis are presented in Tables 9 and 10. Table 9 summarizes the data required for the F-tests on significant differences between means of individual problems. Table 10 shows the probabilities of these differences. Table 11 is included to show the comparison of findings obtained from an analysis of covariance and those obtained from an analysis of variance on each criterion variable alone.

The findings indicate that the control group performed significantly better than the project group on Problem #4 (Quadratic Functions), therefore, the hypothesis was rejected on Problem #4. An inspection of the cell means reveals that the project group mean scores on Problem #1 (Radicals) and Problem #2 (Polynomials) were greater than those of the control group, but the differences were not statistically significant. The control group mean scores were greater than those of the project group on Problem #3 (Systems of Equations), Problem #4 (Quadratic Functions), and Problem #5 (Trigonometry), but only on Problem #4 was the difference statistically significant. The results obtained from an analysis of covariance are similar to those obtained from an analysis of variance.

TABLE 9

Summary of Analysis

of Individual Problem on Problem Solving Test

Problem	Group	Number	Mean	Variance	Math 10 Mean	Homogeneity of Variance
#1	P	41	1.68	2.37	66.51	x^2 = 1.211;
Radicals	C	43	1.58	1.68	65.60	p = 0.271
#2	P	46	3.57	3.27	65.48	x^2 = 0.023;
Polynomials	C	49	3.49	3.13	62.84	p = 0.880
#3	P	46	2.61	3.84	65.80	x^2 = 1.519;
Sys. of Eq.	C	54	2.89	5.50	63.39	p = 0.218
#4	P	32	2.13	4.11	68.16	x^2 = 0.280;
Quad. Fnt.	C	59	4.00	3.48	66.49	p = 0.600
#5	P	31	4.58	4.52	68.87	x^2 = 0.738;
Trigonometry	C	43	4.95	3.38	66.88	p = 0.390

TABLE 10

Analysis of Covariance

of Individual Problem on Problem Solving Test

Problem	Source of Variation	Sum of Squares	D.F.	Mean Square	F Ratio	p
#1	Groups	0.17	1	0.17	0.08	0.771
Radicals	Math 10 (Cov.)	1.59	1	1.59	0.79	0.377
	Errors	163.75	81	2.02		
#2	Groups	0.01	1	0.01	0.00	0.962
Polynomials	Math 10 (Cov.)	5.35	1	5.35	1.68	0.197
	Errors	292.18	92	3.18		
#3	Groups	5.32	1	5.32	1.29	0.259
Systems of	Math 10 (Cov.)	63.35	1	63.35	15.33	0.000
Equations	Errors	400.91	97	4.13		
#4	Groups	79.95	1	79.95	24.16	0.000
Quadratic	Math 10 (Cov.)	38.26	1	38.26	11.56	0.001
Functions	Errors	291.25	88	3.31		
#5	Groups	2.49	1	2.49	0.78	0.381
Trigono-	Math 10 (Cov.)	49.30	1	49.30	15.35	0.000
metry	Errors	228.08	71	3.21		

TABLE 11

Analysis of Variance

of Individual Problem on Problem Solving Test

Problem	Source of Variation	Sum of Squares	D.F.	Mean Square	F Ratio	P
#1	Groups	0.22	1	0.22	0.11	0.744
Radicals	Errors	165.34	82	2.02		
#2	Groups	0.13	1	0.13	0.04	0.840
Polynomials	Errors	297.55	93	3.20		
#3	Groups	1.94	1	1.94	0.41	0.523
Sys. of Eq.	Errors	464.29	98	4.74		
#4	Groups	72.93	1	72.93	19.70	0.000
Quad. Fnt.	Errors	329.50	89	3.70		
#5	Groups	2.50	1	2.50	0.65	0.423
Trigonometry	Errors	277.45	72	3.85		

Discussion

The findings indicate that the control group performed significantly better than the project group on Problem #4 (Quadratic Functions), but no significant differences were found on the other problems. Although the project group did more programming in Quadratic Functions than the others, rather than improve problem solving performance, such activity seems to have detracted from performance in this area. In the other four areas there were no significant differences. These findings do not indicate that the IPM Elective enhances student performance on problem solving tasks. In particular even in areas where students program three or four solutions, problem solving gains are not realized in the IPM Elective.

III. Attitude to Mathematics

Hypothesis 5

There are no significant differences between the project and control group mean scores on Evaluation, Usefulness, and Difficulty Scales as measured by the School Subjects Attitude Scales.

Testing Procedure and Findings

Hypothesis 5 was tested using one-way analysis of variance (with unequal sample sizes) (Ferguson, 1976). The analyses are presented in Tables 12 and 13.

The summary of data relevant to the analysis are shown in Table 12. For each attitude scale, the mean score of the project group was greater than that of the control group. Table 13 shows the probabilities that these differences occurred by chance to be 0.241, 0.005, and 0.010 for the Evaluation, Usefulness, and Difficulty Scales, respectively. For the last two scales the probabilities are less than the significance level of 0.05 adopted for the study. Hypothesis 5 was therefore rejected for those two attitude scales. The mean scores of the project group on the School Subjects

Attitude Scales were significantly greater than those of the control group on the Usefulness and Difficulty Scales. The project group thought that mathematics was more useful, more important, and easy.

TABLE 12

Summary of Analysis on School Subjects Attitude Scales

Scale	Group	Number	Mean	Variance	S.D.	Homogeneity of Variance
Evaluation	P	75	25.19	48.02	6.93	x^2 = 0.073;
	C	85	23.87	51.04	7.14	p = 0.787
Usefulness	P	75	34.71	37.10	6.09	x^2 = 1.585;
	C	85	31.72	49.40	7.03	p = 0.208
Difficulty	P	75	20.96	37.90	6.16	x^2 = 0.001;
	C	85	18.41	37.63	6.13	p = 0.978

TABLE 13

Analysis of Variance on School Subjects Attitude Scales

Scale	Source of Variation	Sum of Squares	D.F.	Mean Square	F Ratio	p
Evaluation	Groups	68.75	1	68.75	1.39	0.241
	Error	7840.87	158	49.63		
Usefulness	Groups	355.66	1	355.66	8.15	0.005
	Error	6894.68	158	43.64		
Difficulty	Groups	258.42	1	258.42	6.84	0.010
	Error	5965.38	158	37.76		

Discussion

Although the control group's achievement mean score was significantly greater than that of the project group at every ability level, the mean score for each attitude scale of the project group was greater than that of the control group with significant differences on the Usefulness and Difficulty scales. Obviously, the IPM Elective had a positive effect on students' feelings toward mathematics. Programming exercises seems to be positively associated with the students' attitudes to mathematics.

General Discussion on the Product Evaluation

The findings of this project do not generally support the IPM Elective in enhancing student achievement or problem solving capabilities. However, in areas where a significant amount of work is done better results can be expected. It also appears that the activity of writing computer programs to mathematics exercises is positively associated with the students' attitudes to mathematics. Such a claim is made because, although the project students responded significantly less favorably to their instructional settings on the Teaching Scale than the control students did to theirs, they still had more positive attitudes to mathematics especially in the areas of usefulness and difficulty. It is also interesting to note that although the project group scored significantly lower in achievement and in one area in problem solving, they indicated that they found mathematics easy. The findings seem to show that the programming activity of the IPM Elective itself enhanced the students' attitude and motivation to learn mathematics. Or perhaps compared to the challenge of programming, mathematics seems easy. In any case the IPM Elective significantly changed the students' perception of mathematics.

CHAPTER VII

RESULTS, INTERPRETATIONS, AND IMPLICATIONS

I. Purpose and Design of the Study

The purpose of the study was to investigate the implementation of an elective unit called Integrating Programming into Mathematics (IPM) in the Math 20 program which would teach students to program, in BASIC, solutions to typical end-of-chapter exercises, and to determine the benefits that were gained by the students in this particular type of learning situation. Because the study focused on curriculum development and implementation of the IPM Elective, students' reactions to the elective and attitude to mathematics were also studied.

The classes which participated in the IPM program were Math 20 classes. In this design, five classes of students were assigned to the IPM program.

The curriculum development took place in the first semester. Five Math 20 teachers took part in the IPM program in both semesters. Semester one saw an agreed-upon course being followed by the five teachers. Evaluation in the second semester took the form of a comparison with the group of students in the regular Math 20 classes in three other senior high schools in the same school district.

II. Summary of Results

The results of the project were initially reported in Chapter IV, V, and VI. They are presented in the present section in the context of summary of results investigated by the project to provide a basis for the interpretations drawn by the project.

Curriculum Development and Implementation

Five Math 20 teachers and a curriculum consultant helped the researcher to develop two manuals for utilizing in the IPM program. The development took place in the first semester (September, 1983-January, 1984), and the implementation took place in the second semester (February-June, 1984).

Curriculum Development

1. Student Manual for BASIC Language

The manual was set up in five lessons with practice questions to conclude each lesson. These lessons were offered for a one-hour period per week. It was recommended that the first three lessons be conducted in the classroom rather than in the computer lab. The first three lessons contain five key ideas; components of a computer, immediate and deferred modes, BASIC operations and program statements, system commands, and analyzing and modifying a program. The fourth lesson contains the iterative procedure statements (IF-THEN and FOR-NEXT) for programming. The use of the disk and disk drive is included in the fifth lesson.

2. Student Exercises for Programming in BASIC - Math 20

The manual is a collection of typical end-of-chapter exercises, the solutions of which could be programmed in elementary BASIC language. The exercises were arranged according to the Alberta Math 20 curriculum. In general, easier exercises were given first within a section. Within each of the eleven topics of the Math 20 curriculum, three types of exercises were given; data generating exercises such as solving specific equations (procedures) with given numerical coefficients, applying formulae exercises such as solving for a form of an equation (procedure), and more complex exercises such as solving for a pre-set range of variables or numerical coefficients. The criterion for a good program was whether it gave a correct answer, in some acceptable format. The students were then encouraged to verify their own programs by substituting values which gave known answers.

Curriculum Observations

Although curriculum observation was not formally a part of this project, some informal observations should be mentioned.

1. The five teachers tended to offer the IPM Elective differently, especially in such areas as teaching BASIC, assignment of exercises, amount of guidance in programming, grading of programs, and student helpers.

2. Some teachers found it necessary to assign computer exercises in the regular mathematics class hour previous to the computer lab hour. Students needed to be encouraged to think programs through before coming to the lab period.

3. Some teachers made effective use of students who had good computer backgrounds as student helpers both in the classroom and the computer lab.

4. Class frustration levels grew during the fourth and fifth lessons when students began to work seriously at entering and saving programs.

5. Having two or more types of computers in the computer lab added greatly to the confusion and frustration of lab periods. Teachers must be completely familiar with the operation of the computer installation in the computer lab.

1. Students' Views on the IPM Elective

How do students react to learning mathematics in the IPM Elective? What are their reactions to their instructional setting?

The analysis of the responses obtained from the first 23 items of the Student Opinionnaire ((A) and (B)) indicated that the students in the IPM program needed more help, more attention, and more explanation from their teachers.

The IPM students' reactions to computer programming and mathematics learning items indicated the IPM students felt that programming was an interesting part of mathematics and it should be more than one hour per week in order to have enough time to work on exercises. The students also mentioned that there should have been more in-class instruction. The majority of the IPM students preferred to do mathematics with programming exercises, but less than one-half of them thought that programming exercises helped them to use mathematical knowledge to solve problems.

The students' reactions in the regular classes to the possibility of their having experiences in computer programming in mathematics revealed that the majority of these students would like to have experience in computer programming. They felt that a mathematics course should involve computer programming.

2. Student Production

How effectively do students engage in the IPM Elective? On the average, the IPM students did only one exercise per session. It does appear that the good mathematics students did not do much programming, but the average students did more programming than the other two groups. In any case, the actual number of programs written was not of the magnitude one would have thought; very low production actually occurred.

Product Evaluation

1. Achievement in Mathematics

Does the IPM Elective contribute to or detract from achievement? How well do students learn mathematics?

In the areas where the IPM students did a significant amount of programming an increase in students' achievement (or at least no decrease) in that area could be expected. The minimum of three programming exercises per topic should be completed by the students in order to improve or maintain their own achievement. On the other hand, in the IPM Elective in areas where a significant amount of work is not done, one would expect a decrease in achievement scores. The

IPM Elective seems to negatively affect achievement especially in those areas where students do less than three programming exercises. The uniformity of achievement differences across ability levels suggests that one hour per week spent on the IPM Elective which meant one hour per week less in the regular mathematics course had a negative affect on all students.

2. Problem Solving

Does programming computer solutions to typical end-of-chapter exercises in grade eleven mathematics (Math 20) improve students' problem solving performance? The results do not indicate that the IPM Elective enhances student performance on problem solving tasks. In particular, even in areas where students programmed three to f[illegible] solutions, problem solving gains were not realized in the IPM Elective.

3. Attitude to Mathematics

What effect does the learning of mathematics in the IPM Elective have on the attitude of students to the subject? The findings indicate that the students in the IPM Elective find mathematics easy and useful. The findings seem to show that the programming activity of the IPM Elective enhances the students' attitude and motivation to learn mathematics.

III. Interpretations

The Meaning to the Students

The IPM students preferred to do mathematics with programming. There are a number of reasons for this preference. One possible reason is because of the popularity of the microcomputer in society. Another possible reason is that they found working with the computer to be a challenge. In the programming mode, they have to design an algorithm for the computer to handle (or perform) the instruction that they provide to the computers. This induces the students to view programming as an important activity, even though they did not see the relationship between programming and mathematics learning. They viewed the Elective as a separate course from the regular mathematics course and felt that it was not so important (in terms of grades) to spend time on the Elective. They were frustrated by not having enough help from the teachers. Therefore, the students lost their commitment to the Elective in terms of programming activity even though they still saw it as an important activity. Lost commitment is reflected in low productivity. Perhaps, if the teachers would have been able to show the connection between mathematics learning and programming or could have integrated it more into the regular mathematics course, then students might have seen it as more important. Being a part of a mathematics course would guarantee the possibility of a significant amount of programming activity. As it was the students (and even teachers) probably treated the IPM Elective as an extra hour to the mathematics program. The range of accomplishment was much broader in the Elective than in the regular mathematics course. Only students who were quite good at programming were able to get a significant amount of work done in the hour. Perhaps the students who did well were students who had a background in programming.

The results of this project do not generally support the IPM Elective in enhancing student achievement. However, in areas where a significant amount of programming is done there is some evidence of better achievement.

In the problem solving area, the result of this study holds little hope. The results seem to show that the IPM Elective is rather hard to implement and within the context of senior high school mathematics it is difficult to get enough programming activity to improve problem solving performance.

In the Elective, as the students gained experience in programming mathematics exercises, they seemed to see the application of mathematics and programming as a way to apply mathematics. This caused them to view mathematics as a more important, valuable, and necessary subject which resulted in their seeing mathematics as a useful subject.

The IPM students felt that mathematics was easy. There are several possible reasons for this. One of them is perhaps that they compared the mathematics course with the IPM Elective. Other possible reasons are: programming activity in the IPM Elective might help students develop an algorithmic view of mathematics which would be more understandable and more manageable; or they might feel that the five-hour Math 20 (excluding one-hour elective) easier than the six-hour Math 20 (including one-hour elective in regular mathematics program) in which they had to learn more concepts and do more work; or they might feel the five-hour Math 20 plus one more relaxing hour of the IPM Elective was easier than the five-hour Math 20 plus one-hour elective in another topic in difficult mathematics. However, the programming activity in the IPM Elective seems to be associated with the students' positive feeling toward mathematics.

The Teacher Variables

Most of the interpretation about teachers comes from the Process Monitoring. However, because the researcher spent a considerable amount of time in the lab, some of his informal observations are stated below.

Efficiency. The teachers did not adhere closely to suggested guidelines, especially in the first five lessons in BASIC. Teachers should teach the Elective as a more integral part of the regular mathematics program. They should assign homework regularly and get the students to read the BASIC Manual.

Major Undertaking. The teachers found the IPM Elective to be a major undertaking. The Elective needed carefully thought-out teaching strategies such as the use of problem sheets to guide the students' learning of programming. Teachers should keep in mind that students should learn to write the computer programs but should not play with the computers. The computer lab generates student excitement and teachers must try harder to capitalize on this excitement. The assignments should be followed up more regularly by the teachers.

The Computer Concept. In the IPM Elective, the computer is used as a tool not as a focus of the study. In the project situation, the computers were not fully operational. Hours were spent getting the tool to work. Computer malfunctions add frustration to a demanding programming task. Teachers' familiarity with the computer lab operation is essential.

Teacher variables play an important role in the process of curriculum development and implementation. Teachers' knowledge and preparation shapes the everyday activity and learning atmosphere in their classrooms which contributes a great deal to students' learning.

Curriculum Implementation

A general impression is that the degree of curriculum implementation was not high in this project. The researcher would like to express some comments on the process of implementation used by the five teachers in the IPM program;

1. The curriculum that is developed by teacher(s) does not necessarily get used by the teacher(s). The five senior high mathematics teachers helped the researcher develop the two manuals in the first semester, but they did not implement (or use) them very effectively in their classes in the second semester. From the researcher's understanding this is because they were very busy with the regular mathematics course, putting their effort into the regular mathematics course. Perhaps the IPM Elective was a whole new idea for them. Even though they participated in the process of curriculum development their ownership of the curriculum was still not high enough for them to become committed to it.

2. Teachers' previous knowledge and beliefs have a large impact on curriculum implementation. The five teachers had very different backgrounds in BASIC. Some had a good background in BASIC while the others had none. Their knowledge and beliefs in presenting the BASIC lessons crucially affected their approaches to the BASIC lessons which resulted in too much time being spent on the BASIC lessons. Because of this the students had less time for programming mathematics exercises which played an important role for students' mathematics learning in the IPM Elective. Perhaps to have the teachers implement the IPM program more effectively more time may be needed for the teachers to learn the process of implementation.

3. Teachers should be better prepared for the role

change, especially their new role in the computer lab. The researcher experienced a wide range of teachers' capability in handling the students in the computer lab. In the first semester the researcher helped the five teachers in the computer lab one hour a day, five days a week. All teachers were asked to conduct the first three BASIC lessons in the classrooms and before going to the lab they should have done 15 minutes of class-instruction. But all of them preferred to have the students work in the lab from the beginning of the first lesson. Some teachers did the suggested 15 minutes class-instruction before having students work in the lab. The researcher found that once the students got into the lab the teachers could not get all the students' attention when they wanted to explain something that seemed to be a common problem or difficulty for the students in writing programs or operating the machines. They had to supervise the students individually at their stations. Many times before letting the students work with the machines, the teachers had to resort to turning the main power switch off in order to get all of the students' attention when they wanted to explain or teach in the lab.

4. Teacher in-service did not focus enough on raising the teachers' commitment and increasing their understanding of the program. Even though initially teachers were willing to take part in the program, their lack of commitment throughout the program played an important role in curriculum implementation.

5. Instead of being integrated as a unit into the regular mathematics course, the IPM Elective was treated as a separate course from the regular mathematics course. The teachers gave less importance to the Elective than to the mathematics course. Therefore, the teachers did not assign homework and did not assign the students to read the BASIC Manual regularly. They did not even follow up the previously assigned homework. Because of a limited amount of time available, the teachers often let the students continue to the next lesson without finishing the previous lesson. If the teachers would have followed more closely the suggested guidelines for the first five lessons of the BASIC Manual, the remaining time for programming mathematics exercises might have been enough for the students to accomplish more. The students should have been assigned homework in the previous hour so that they could think a program through on a sheet of paper before coming to the computer lab. The fact is the teachers found this difficult to do because they had a lot of work and also a lot of content to be covered in the regular mathematics course.

6. Developing a curriculum separate from the regular mathematics program will probably lead students to think the separate program is less valuable. By attaching the IPM Elective to the regular mathematics course, the students' motivation and commitment to the Elective was affected.

IV. Implications for Practice

The researcher would like to present the implications for mathematics teachers for continuing use of the IPM program. The implications are derived from experiences with the IPM Elective in teaching and learning mathematics and from what is known about how to change educational practices.

1. The IPM program can be offered again the way it was offered in this project, if special attention is paid to:

a. Teacher preparation. Teachers need to be computer literate, with expertise especially in the areas of computer programming and the operation of the computer.

b. Integrating the IPM program into the regular mathematics curriculum. The emphasis should be on curriculum that develops mathematics learning through programming rather than to introduce programming.

c. Regularly assigning and following up on homework.

d. Assigning the students to read and use the BASIC Manual as a reference.

e. Grading the students' activity in the IPM program as is done in the regular mathematics course.

2. The IPM program can be changed to be fully integrated into the mathematics program. One possible idea is the first five BASIC lessons could be written into mathematics textbooks. Learning mathematics through programming must be a part of the textbooks used by teachers and students. Mathematics teachers usually have neither the

time nor the means to collect and sort through resource materials and programs related to computers and mathematics.

3. With the above two possible practices, the teachers must keep in mind that the primary purpose of the IPM program is to develop mathematical concepts and skills through programming experiences. First and foremost, a programming enhanced mathematics curriculum must be a mathematics curriculum. As such, the computer becomes a tool for teaching and learning mathematics rather than an object of instruction.

V. Implications for Research

The researcher would like to suggest some ideas for pursuing research in the area of curriculum development and implementation which are drawn from experiences in the present project.

1. Since the teachers are the most important contributors to curriculum development in our schools, further curriculum study with teachers in actual classroom situations is recommended. Ideas and approaches should be carefully prepared and conducted in the in-service training of teachers. A better implementation procedure should also be developed.

2. Information processing research that particularly looks at the effect of programming mathematics exercises on students' approach to solving mathematics problems is

recommended.

3. Another area of research is to develop new curriculum approaches which would incoporate mathematics curriculum and programming activity. One possible idea is to integrate programming in BASIC into the first unit of a mathematics course and allow students to learn mathematics in later units through programming.

4. One conclusion of the study was that the implementation process was not effective. In particular, the effects of teachers' and students' reactions to the implementation was poorly understood. A study that has the same design as this study could be undertaken paying more attention to teacher and student variables in implementation process.

5. One hypothesis suggested by the study is that the real impact of computers on mathemmatics is an attitudinal one. A study on programming mathematics exercises to enhance students' attitude to mathematics is also recommended.

BIBLIOGRAPHY

Alberta Education. Curriculum guide for Mathematics 10, 20, 30. Edmonton, Alberta: Department of Education, 1983.

Andreoli, T. F. The effects of a programmed course in computer programming with different feedback procedures on mathematical reasoning ability (Doctoral dissertation, University of Connecticut, 1976). Dissertation Abstracts International, 1976, 37, 780A.

Basil, G. J. The effects of writing computer programs on achievement and attitude in elementary calculus (Doctoral dissertation, University of Pittsburgh, 1974). Dissertation Abstracts International, 1974, 35, 2114A-2115A.

Bell, F. H. Can computer really improve school mathematics? Mathematics Teacher, 1978, 71, 428-433.

Billings, K. Developing mathematical concepts with microcomputers activities. Arithmetic Teacher, 1983, 30, 18-19, 57-58.

Bozeman, W. C., Altachuler, H., Dorazio, R., and Spuck, D. W. Layman's guide to the use of computer in education. Association for Educational Data System, 1982.

Coan, J. S. Basic Apple BASIC. Rochelle Park, New Jersey: Hayden Book Company, Inc., 1928.

Coburn, P., Kelman, P., Roberts, N., Snyder, T., Watt, D. H., and Weiner, C. Practical guide to computer in education. Massachusetts: Addison-Wesley Publishing Co., 1978.

Deloatch, S. J. A comparative study of the use of computer programming activities in an introductory college mathematics course for disadvantaged students (Doctoral dissertation, Indiana University, 1977). Dissertation Abstracts International, 1978, 38, 6585A.

Division of Educational Research Services. Info 17: summary of statistical computer programs. (Available from DERS, Faculty of Education, The University of Alberta, Room 3-104 Education Centre North, Edmonton, Alberta, Canada T6G 2G5), January 1984.

Ebos, F., and Tuck, B. Math is/5. Toronto: Nelson/Canada, 1980.

Ferguson, G. A.. Statistical analysis in psychology and education. Toronto: McGraw Hill Book Co., 1976.

Foster, T. E. The effect of computer programming experiences on student problem solving behaviors in 8th grade mathematics (Doctoral dissertation, University of Wisconsin, 1972). Dissertation Abstracts International, 1973, 33, 4239A-4240A.

Fryer, K. D., Dunkley, R. G., Elliott, H. A., Hill, N. J., and Mackay, R. J. Holt Mathematics 5. Ontario: Holt, Rinehart and Winston of Canada, Ltd., 1980.

Fullan, M. Conceptualizing problems of curriculum implementation. In W. Werner (Ed.), Curriculum Canada. Vancouver, British Columbia: Centre for the Study of Curriculum and Instruction, University of British Columbia, 1979.

Fullan, M. Curriculum implementation: a resource booklet. Toronto: Ontario Institute for Studies in Education, 1981.

Fullan, M. The meaning of educational change. Toronto: OISE Press, 1982.

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

Glaser, R., and Lesgold, A. Psychological and pedagogical issues for computer technologies in education. Paper presented at CBE Research Conference, June 3-4, 1982.

Goodfellow, D. C. Apple II BASIC. Blue Ridge Summit, PA.: Tab Books Inc., 1983.

Greeno, J. G. Understanding and procedural knowledge in mathematics instruction. Educational Psychologist, 1978, 12, 262-283.

Greeno, J. G. Psychology of learning, 1960-1980: one participant's observations. Technical report number 5. Pittsburgh: Learning Research and Development Center, University of Pittsburgh, 1980.

Greenes, C. The computer in mathematics education. Mathematics Teacher, 1981, 74, 588-589.

Hatfield, L. L. Computer-assisted mathematics: an investigation of the computer used as a tool to learn mathematics (Doctoral dissertation, University of Minnesota, 1969). Dissertation Abstracts International, 1970, 30, 4329A-4330A.

Hatfield, L. L. Computers in mathematics education. Paper presented at the National Council of Teachers of Mathematics Annual Meeting, Minneapolis, Minnesota, April, 1969.

Hatfield, L. L. Teaching mathematics with microcomputers: junior high school. Arithmetic Teacher, 1983, 30, 44-45, 68-69.

Haylock, D. W. Understanding in mathematics: making connections. Mathematics Teaching, 1982, 98, 52-54.

Hill, S. A. The microcomputer in the instructional program. Arithmetic Teacher, 1983, 30, 14-15, 54-55.

Hoffman, I. J. Effect of computer application on generalization skills and achievement in second year algebra course (Doctoral dissertation, University of Denver, 1971). Dissertation Abstracts International, 1971, 32, 2856B-2857B.

Howson, G., Keitel, C., and Kilpatrick, J. Curriculum development in mathematics. Cambridge, London: Cambridge University Press, 1982.

Inman, D., and Clude, D. Computer applications. Mathematics Teacher, 1981, 74, 618-620.

Johnson, D. A. Attitude in the mathematics classroom. School Science and Mathematics, 1957, 57, 113-120.

Johnson, D. A., and Rising, G. R. Guideline for teaching mathematics. Belmont, California: Wadsworth Publishing Company, 1967.

Jurick, R. R. Computer-oriented instructional system for teaching analytic geometry (Doctoral dissertation, Ohio State University, 1972). Dissertation Abstracts International, 1972, 33B, 1667-1668.

Katz, S. M. A comparision of the effects of two computer augmented methods of instruction with traditional methods upon achievement of algebra two students in a comprehensive high school (Doctoral dissertation, Temple University, 1971). Dissertation Abstarcts International, 1971, 32A, 1188-1189.

Kelly, A. V. The curriculum: theory and practice. London: Harper and Rows Ltd., 1982.

Kelly, B., Alexander, B., and Atkinson, P. Mathematics 10. Don Mills, Ontario: Addison-Wesley Publishing Co., 1982.

Kemeny, J. The role of computers and their applications in the teaching of mathematics. In H. A. Fehr (Ed.), Needed research in mathematics education. New York: Columbia Teachers College Press, 1966.

Kemeny, J., and Kurtz, T. E. Dartmouth time-sharing. Science, 1968, 162, 223-228.

Kieren, T. E. The computer as a teaching aid for eleventh grade mathematics: a comparison study. Unpublished doctoral dissertation, University of Minnesota, 1968.

Krulik, S., and Weise, I. B. Teaching secondary school mathematics. Toronto: W. B. Saunders Company, 1975.

Krull, S. M. An analysis of the effects of learning to program a computer on student math performance and attitude toward school (Doctoral dissertation, St. Louis University, 1979). Dissertation Abstracts International, 1980, 40, 5711A.

Leuhrmann, A. Should the computer teach the students or vice versa? In R. T. Taylor (Ed.), The computer in the school: tutor, tool, tutee. New York: Teachers College Press, 1981.

Lipson, J. I. Technology in science education: the next 10 years. Computer, 1980, 3, 23-28.

Mandelbaum, J. A study of the effects on achievement and attitude, of the use of the computer as a problem solving tool with low performing tenth grade students (Doctoral dissertation, Temple University, 1973). Dissertation Abstracts International, 1974, 34, 3700A.

Mayer, R. E. Instructional variables in meaningful learning of computer programming. Paper presented at the Annual Meeting of the American Educational Research Association, Washington, D. C., March 30-April 3, 1975.

McIsaac, D. N. Impact of personal computing on education. Association for Educational Data System Journal, Fall 1979, 7-15.

Milner, S. D. The effects of teaching computer programming on performance in mathematics (Doctoral dissertation, University of Pittsburgh, 1972). Dissertation Abstracts International, 1973, 33, 4183A-4184A.

Mulaik, S. A. The foundations of factor analysis. New York: McGraw Hill, 1972.

National Council of Teachers of Mathematics. An agenda for action: recommendations for school mathematics of the 1980s. Reston, Va.: The Council, 1980.

Nicholls, A., and Nicholls, S. H. Developing a curriculum: a practical guide. London: Grorge Allen and Unwin Ltd., 1972.

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

Nyberg, V. R., and Clarke, S. C. T. Technical report on the school subjects attitude scales. Edmonton, Alberta: Alberta Education, 1979.

Osborne, A., and Donahue, C. S. PET/CBM personal computer guide. Berkeley, California: OSBORNE/McGraw-Hill, 1980.

Papert, S. Computer and computer cultures. Creative Computing, 1981, 7, 82-92.

Papert, S. Mindstorms: children, computers and powerful ideas. New York: Basic Books, Inc., 1980.

Payne, C. I. The effects of computer mathematics instruction on achievement, problem solving, and attitudes in a public high school (Doctoral dissertation, Indiana University, 1980). Dissertation Abstracts International, 1980, 40, 3850A.

Posamentier, A. S., and Stepelman, J. Teaching secondary school mathematics. Columbus, Ohio: Charles E. Merrill Publishing Co., 1981.

Resnick, L. B., and Ford, W. W. The psychology of mathematics for instruction. Hillsdale, N. J.: L. Erlbaum Association, 1981.

Reyes, L. H. Attitudes and mathematics. In M. Montgomery (Ed.), Selected issues in mathematics education. Berkeley, California: McCutchin Publishing Corp., 1980.

Ronan, F. D. Study of the effectiveness of a computer when used as a taching and learning tool in high school mathematics (Doctoral dissertation, University of Michigan, 1971). Dissertation Abstracts International, 1971, 32, 1264A-1265A.

Sadowski, B. R. A model for preparing teachers to teach with the microcomputer. Arithmetic Teacher, 1983, 30, 24-25, 62-63.

Saunders, J. What are the real problems involved in getting computers into the high school? Mathematics Teacher, 1978, 71, 443-447.

Scheffe, H. The analysis of variance. New York: John Wiley, 1964.

Shumway, R. J. One point of view: let kids write programs. Arithmetic Teacher, 1983, 30, 2, 56-58.

Sigurdson, S. E. Computer programming in senior high school mathematics. Project proposed to Alberta Education, 1983.

Smith, S. Microcomputers in the middle school. In V. P. Hansen and M. J. Zweng (Eds.), Computers in mathematics education: 1984 yearbook. Reston, Virginia: National Council of Teachers of Mathematics, 1984.

Soloway, E., Lochhead, J., and Clement, J. Does computer programming enhance problem solving ability?: some positive evidence on algebra word problems. Paper reported to the Joint National Institute of Education, 1981.

Taylor, R. P. Programming primer. Don Mills, Ontario: Addison-Wesley Publishing Co., 1982.

Tobin, C. D. Developing computer Literacy. Arithmetic Teacher, 1983, 30, 22-23, 60.

Winer, B. J. Statistical principles in experimental design. New York: McGraw Hill, 1962.

Yunker, L. E., Vannatta, G. D., and Crosswhite, F. J. Advanced mathematical concepts. Columbus, Ohio: Charles E. Merrill Publishing Co., 1981.

APPENDIX I

STUDENT MANUAL FOR BASIC LANGUAGE

WITH

ANSWER KEY

Student Manual for BASIC Language

Using the Apple Computer

PAGE

LESSON 1
1.1 Computer Elements....144
1.2 Function Keys....145
1.3 Immediate and Deferred Modes....146
1.4 Utility Commands....147
1.5 PRINT Statement....147
1.6 BASIC Operations....148
1.7 E Notation....149
Exercise....150

LESSON 2
2.1 Review of Lesson 1....152
2.2 Computer Program....152
2.3 System Commands....152
2.4 Variables....155
2.5 LET Statement....155
2.6 GOTO Statement....157
Exercise....157

LESSON 3
3.1 Review of Lesson 2....159
3.2 INPUT Statement....159
3.3 Analyzing Programs....160
3.4 Modifying (Re-writing) a Program....161
Exercise....162

LESSON 4
4.1 Review of Lesson 3....164
4.2 IF-THEN Statement....164
4.3 FOR-NEXT Statement....165
Exercise....167

LESSON 5
5.1 Review of Lesson 4....169
5.2 System Commands to Communicate to the Disk....169
5.3 Using a Printer....172
Exercise....172

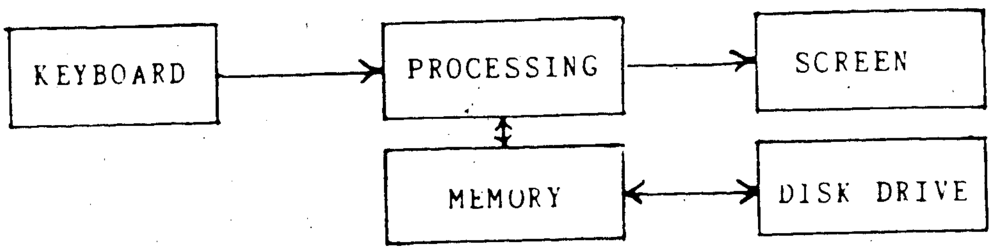
Flow Charting....173

LESSON 1

BASIC, an abbreviation for Beginner's All-purpose Symbolic Instruction Code, is a commonly used computer language. Different computers use different versions of BASIC.

1.1 COMPUTER ELEMENTS

The computer elements are



KEYBOARD provides information to the processing unit.

SCREEN (or printer) displays the processed information or data from the processing unit.

DISK DRIVE (or cassette tape) is used to store information you have in a computer memory onto a disk (or magnetic tape).

PROCESSING occurs in the microprocessing unit (MPU). The MPU obtains a command from Memory and then executes it. Besides adding, subtracting, multiplying, and dividing, the MPU can move information from one spot in the computer to another. In other words, not only can it work with numbers but with letters and symbols as well.

MEMORY is where the MPU finds its instructions and the data it is to work with. There are two main types of memory - Read Only

Memory (ROM) and Random Access Memory (RAM).

ROM: A memory that has a program permanently written into it and cannot be modified by the user. The computer can never write any information into this memory; it is a "read only" memory. ROM is pre-programmed to understand the language (commands) which allow the computer to carry out certain functions. The contents of ROM are not lost when electrical power goes off for any reason.

RAM: A memory system that consists of addresses. These addresses may be "written to or read from" during programming or program execution. Data is placed in specific locations. BASIC statements go into these specific locations when you write a program and are retrieved when you execute the program. Everything in RAM vanishes if electrical power is lost for any reason.

1.2 FUNCTION KEYS

1). CTRL (Control) KEY: TO stop execution of a program hold down the CTRL key and type C. If this doesn't work press CTR key then RESET key.

2). ARROW KEYS:

Right Arrow Key (⟶): The right arrow key moves the cursor one space to the right and rereads any character it has passed over back into computer memory.

Left Arrow Key (⟵): The left arrow key moves the cursor one space to the left and erases each character it has passed over from computer memory, even though the character still appears on the screen.

3). RETURN KEY: The RETURN key must always be pressed after each statement or command is typed. This key transmits the typed statement or command to the computer memory.

1.3 IMMEDIATE AND DEFERRED MODES

Statements may be entered into the microcomputer in either Immediate Mode or Deferred Mode.

1). IMMEDIATE MODE: When a statement is typed into the computer and is executed as soon as the RETURN key is pressed, the computer is said to be in IMMEDIATE MODE. The only way to repeat the statement is to re-type it into the computer. Using the microcomputer in this manner is very similar to using a calculator.

An Example of IMMEDIATE MODE:

```
PRINT 5 + 3 (hit RETURN key)
8
```

2). DEFERRED MODE: Programming occurs in DEFERRED MODE, making it the most frequently used mode. When statements are stored in memory as they are typed in and are executed only when the command RUN is typed, the computer is in DEFERRED MODE. The distinguishing difference between IMMEDIATE and DEFERRED MODES is that statements in DEFERRED MODE are preceded by a line number.

An Example of DEFERRED MODE:

```
10 PRINT"WHAT IS YOUR NAME?" (hit RETURN key)
RUN (hit RETURN key)
WHAT IS YOUR NAME?
```

.4 UTILITY COMMANDS

These are basically Immediate Mode commands that are often useful in the execution of a program.

1). HOME: This command clears the display screen and positions the cursor to the upper left-hand corner of the screen. It does not erase a program from memory. In programming, the HOME command is used to start any new display on the screen.

2). NEW: Because the computer stores programs in its memory, you must specifically instruct it to erase an old program from memory before you type in a new program. Do this by typing in NEW then hit RETURN key. If you forget to type NEW, your new program will be mixed in with the old program still in memory e command NEW erases the contents of RAM. It does not clear the reen nor does it erase anything that is on your disk.

1.5 PRINT STATEMENT

If you want the computer to display something on the screen during program execution, you must issue a PRINT command as part of the program statement. To define what you want the computer to print, type in the PRINT command and enclose whatever you want displayed in quotations (","). If you want to print the result of a calculation do not use quotes (see line 10 below). You can abbreviate the PRINT command by typing in a question mark (?).

Examples of PRINT:

```
10 LET A=2
20 LET B=4
30 LET C=6
40 PRINT"PROBLEM 2"
50 PRINT
60 PRINT A,B,C
70 PRINT A/2,B/2,C/2
80 PRINT
90 PRINT A;B;C
100 PRINT A+1;B+1;C+1
110 END
RUN
PROBLEM 2

2              4              6
1              2              3

246
357
```

A note for PRINT statement:

1). Anything placed between quotation marks will be printed exactly as typed.

2). A comma tells the computer to divide the line on the screen so that variables or messages are printed 15 spaces apart.

3). A semicolon tells the computer to print the line on the screen with no space between each variable or message printed.

4). A PRINT statement with no messages will leave one blank line in the output.

1.6 BASIC OPERATIONS

BASIC Operations: In BASIC, numbers are operated on by arithmetic operations. These operations must be in a form that understood by the computer. A list of operations and corresponding notation in BASIC is given below.

Operation	BASIC Notation
Addition	+
Subtraction	-
Multiplication	*
Division	/
Raising to a power	^

In BASIC an operations symbol cannot be omitted. For example, the product of A and B must be written A*B, not AB. The order of operations in BASIC is the same as in Algebra.

The following examples illustrate the correct notation of some expressions and also the value of each expression.

Expression	BASIC Notation	Value of Expression
2(8+2)	2*(8+2)	20
$(3-1.5)^2$	(3-1.5)^2	2.25
$\frac{4\times9+4}{15-5^2}$	(4*9+4)/(15-5^2)	-4

1.7 E NOTATION

E notation is the computer equivalent of scientific notation. The output of a program in BASIC generally contains a maximum of nine significant digits. When there would be more than nine significant digits, the computer will round off and shift to E notation. Three examples of E notation are given below.

Number	Scientific Notation	E Notation
580,000,000,000	5.8×10^{11}	5.8E + 11
4,280,920,616	$4.280920616 \times 10^{9}$	4.28092E + 09
0.0000264280	2.64280×10^{-5}	2.64280E - 05

EXERCISE

1. State the commands required by your computer to:

   (a) Clear the computer memory.

   (b) Clear the screen.

2. On your computer system, what operations are required to:

   (a) Transmit a typed statement or command to the computer memory?

   (b) Correct a character in a line before the RETURN key is pressed?

   (c) Move cursor left one character?

3. List the five arithmetic operations used in BASIC and their BASIC symbols.

4. Each of the following expressions is incorrectly written in BASIC. Rewrite each expression using the correct BASIC notation.

   (a) 5X (b) 184 ÷ 2 (c) 2(X+Y)

   (d) $\sqrt{3\frac{1}{2}}$*Y (e) 2 + X^2 + 3X

5. State whether each of the following statements is in Immediate Mode or Deferred Mode.

   (a)
```
10 PRINT "(10+5)/3 = ";(10+5)/3 (hit RETURN key)
RUN (hit RETURN key)
(10+5)/3 = 5
```

(b) PRINT "20-15 = ";20-15 (hit RETURN key)
    20-15 = 5

6. Predict output of the following program.

```
10 PRINT"EXERCISE #1"
20 PRINT 2,4,6
30 PRINT"1";"3";"5"
RUN
```

7. Write each of the following numbers using E notation.

(a) 7,600,000,000   (b) 0.001278123   (c) 0.0000000005

LESSON 2

2.1 REVIEW OF LESSON 1

2.2 COMPUTER PROGRAM

A computer program consists of a series of instructions required to solve a specific problem on a computer. The purpose of a computer program is to put instructions and data into the computer (input), have the computer do the execution, and get the results out of the computer (output).

The standard BASIC language consists of about twenty statement types, such as LET, PRINT, and GOTO. Each statement in a BASIC program has a line number which identifies the line and also specifies the order in which the statements are to be executed by the computer. The program below shows how various statements are used.

10 LET X = 5	- Assigns 5 to be value of X.
20 LET Y = 7	- Assigns 7 to be value of Y.
30 PRINT X*Y,X+Y	- Computes & prints the values of X*Y and X+Y.
40 END	- line 40 tells the computer to terminate the program.

2.3 SYSTEM COMMANDS

1). LIST: To display a program currently stored in memory, type LIST, then press RETURN key. There are two types of LIST commands. Notice that if the LIST command does not have a line number, then it will list everything in the memory.

An Example of LIST:

```
LIST
10 PRINT"4 + 5 = ";4+5
20 PRINT"8 - 2 = ";8-2
30 PRINT"3 * 9 = ";3*9
40 PRINT"10 / 5 = ";10/5
50 END
```

Sectional listing is also possible. The LIST command followed by one line number will result in that line being listed. A LIST command with two line numbers separated by a comma will result in listing two lines and all the lines between them. For example,

```
LIST 10,30
10 PRINT"4 + 5 = ";4+5
20 PRINT"8 - 2 = ";8-2
30 PRINT"3 * 9 = ";3*9
```

This last sectional listing may also be written LIST 10-30.

2). RUN: This command instructs the computer to execute your program, starting at the lowest line number and terminating when either an END statement, or the highest line number is reached. Type RUN, press RETURN key. Notice that RUN command does not have a line number.

Examples of RUN:

```
10 PRINT"HELLO"
20 PRINT"HOW ARE YOU?"
30 END
RUN
HELLO
HOW ARE YOU?
```

3). END: This command causes a program to cease execution, and returns control to the user.

4). REM: REM stands for remark. A REM statement is put in to remind the programmer of the type of the program. It does not affect the execution of the program. The general form of the REM statement is as follows:

line number REM comment

An Example of REM:

```
10 REM FIND AVERAGE OF THREE TEST SCORES
20 PRINT"TEST SCORES ARE 4, 6, 8"
30 PRINT"AVERAGE = ";(4+6+8)/3
40 REM CALCULATES AND PRINTS AVERAGE
50 END
```

5). DEL: The DEL command is used to delete a line or a group of sequential numbers from a program. A DEL command with two line numbers separated by a comma will result in deleting the two lines and all the lines between them.

An Example of a group of sequential line DELeting:

```
LIST
10 REM ** FIND THE SQUARES OF 10 AND 20 **
20 LET N=10
30 PRINT N,N^2
40 END
DEL 20,30
LIST
10 REM ** FIND THE SQUARES OF 10 AND 20 **
40 END
```

An Example of one line DELeting:

```
DEL 10,10
LIST
40 END
```

Another way of deleting a single line in a program is to simply type the line number of the line you want to delete and press RETURN key.

2.4 VARIABLES

A variable is a symbol that is used in programming to represent a number (or other quantities not discussed here.) It refers to that place in the computer memory where the number is stored. In other words, a variable can be thought of as an address that designates a location in the memory. This location contains a value which may vary as the program is being executed.

In BASIC, a variable may be represented by a single letter such as A, B, C,..., Z, or two letters such as AB, AC, BC,..., or a letter followed by a single digit such as A0, A1, A2,..., B0, B1, B2,..., Z9. The first letter of a variable must always be alphabetic and must be typed without quotation marks (" ").

2.5 LET STATEMENT

Specific values can be assigned to a variable by using a LET statement. Consider each of the following programs.

```
10 LET A = 2
20 LET AB = 8
30 LET A1 = 5
40 ? (A + AB)*A1
50 END
RUN
50
```

This next program assigns Y a value in terms of the variables X, A, and C.

```
10 LET A = 2
20 LET C = 5
30 LET X = 10
40 LET Y = X*A + C
50 PRINT "Y = ";Y
60 END
```

The equals sign tells the computer to assign the value of the expression on the right to the variable on the left. The left side of the equation must have only one variable. Variables may appear on the right side of the equation as above. However, each variable should be assigned a specific value prior to the statement in line 40.

Consider the following programs.

```
10 LET X = 4          - X and Y are assigned as follows.
20 LET Y = 2                 X = 4        Y = 2
30 LET A = 5*Y + 20/X - The value of 5*Y+20/X is then
40 PRINT A              computed and assigned to A in
50 END                  line 30. This value is then
RUN                     printed in line 40.
15                    - Output
```

An Example of LET (with an increment):

```
10 LET X = 1
20 LET X = X + 2
30 PRINT X
40 END
RUN
3
```

The above program shows that the use of the equal sign in programming differs from its use in algebra. (In algebra, X is not [illegible] X+2).

2.6 GOTO STATEMENT

GOTO is the simplest branching statement; it UNCONDITIONALLY causes program execution to branch to anywhere in the program specified by the GOTO statement. This may be anywhere above or beyond the current point of program execution. Once the GOTO statement has been executed, program execution continues sequentially from the line number indicated in the GOTO statement.

Consider the following program.

```
10 LET X = 1
20 LET X = X + 2
30 PRINT X
40 GOTO 20
50 END
RUN
1
3
5
etc.
```

Line 40 introduces a GOTO statement. The computer will return to line 20 each time it comes to line 40. You may RUN the above program on your computer to see more output. The GOTO statement has a general form as follows.

line number <u>GOTO</u> line number

EXERCISE

1. State the system commands required by your computer to:

   (a) Display a program on the screen.

   (b) Execute a program.

   (c) Delete a line from a program.

   (d) Terminate a program.

2. Each of the following programs contains at least one error Find each error and write a correct program.

(a)
```
20 PRINT 6(5+4)
15 END
```

(b)
```
10 PRINT 3^2
20 END
30 RUN
```

(c)
```
LET A=6
LET Y=A*3
PRINT Y
END
```

3. Determine whether each of the following is an acceptable BASIC numeric variable. Write yes or no.

(a) X (b) XY (c) A8

(d) 3X (e) B74 (f) DIF

4. State whether each of the following is an acceptable LET statement. Write yes or no.

(a) 20 LET A1=6 (b) 20 LET 8=X

(c) 20 LET X=2*X (d) 20 LET A+B=Z

(e) 20 LET N=R-1 (f) 20 LET X/2=Z/2

5. Show the output of each of the following programs.

(a)
```
10 LET N=3
20 LET Y=5
30 LET Y=Y+7
40 LET X=Y+N
50 PRINT X
60 END
```

(b)
```
10 LET T=4
20 PRINT T,T^2
30 LET T=T+3
40 PRINT T,T^2
50 LET T=T+1
60 PRINT 3*T
70 END
```

6. Each of the following programs contains at least one error. Find each error and write a correct program.

(a)
```
LET X=7
LET Y=X+6
PRINT X+Y
END
```

(b)
```
10 LET X=Y+1
20 LET Y=2
30 PRINT XY
40 END
```

(c)
```
10 LET X=7
20 LET Y=A*X
30 PRINT Y
40 END
```

==

LESSON 3

==

3.1 REVIEW OF LESSON 2

3.2 INPUT STATEMENT

The INPUT statement is another way by which data may be entered into a computer without having to modify the LET statement after each RUN. The INPUT statement is used to obtain information from the user who inputs information into the computer via the keyboard. INPUT can request values for any combination of variables. Once you have correctly entered the information, the RETURN key must be pressed so that the information may be processed and acted upon by the computer.

An Example of INPUT:

```
-----------------------------------------------------------------
10 PRINT"ENTER VALUE FOR A";     -Identifies value to enter.
20 INPUT A                       -Requests value for A.
30 PRINT"ENTER VALUE FOR B";     -Identifies value to enter.
40 INPUT B                       -Requests value for B.
50 LET C = A*B                   -Computes the value of A*B
60 PRINT"C = ";C                  and assigns to C. In line
70 END                            60, the value of C is
RUN                               printed.
ENTER VALUE FOR A? 12
ENTER VALUE FOR B? 5
C = 60
-----------------------------------------------------------------
```

From the above program, you may rewrite the program as follows.

```
-----------------------------------------------------------------
10 PRINT"ENTER VALUES FOR A,B";-Identifies values to enter
20 INPUT A,B                    -Requests values for A & B.
30 LET C = A*B                  -Computes the value of A*B
40 PRINT"C = ";C                 and assigns to C in line
50 END                           30, the value of C is
RUN                              printed.
ENTER VALUES FOR A,B? 12,5
C = 60
```

3.3 ANALYZING PROGRAMS

Please consider the following programs.

(1)

```
--------------------
10 LET X = 5
20 LET Y = 10*X + 7
30 PRINT Y
40 END
--------------------
```

From the above program at

line 10, the variable X is assigned to have a value of 5.

line 20, the variable Y is assigned to be equal to 10X+7.

line 30, the value of Y is computed then printed.

line 40, END command causes the program to stop execution.

(2)

```
----------------------------------------
10 REM FIND AREA OF A SQUARE,
20 REM GIVEN LENGTH OF SIDE
30 PRINT"GIVE ME THE LENGTH OF A SQUARE"
40 INPUT S
50 PRINT S,S^2
70 END
----------------------------------------
```

From the above program at

line 10 and 20, the REM statements are used to describe the type of the program.

line 30, identifies value to be entered.

line 40, requests value for S (side).

line 50, prints values of length & area of the square,

line 60, tells the computer to restart the program at line 30.

(If you want to stop running of the above program, press CTRL & C or CTRL & RESET keys simultaneously.)

3.4 MODIFYING (RE-WRITING) A PROGRAM

If you want to run the above program (1) many times you could modify it by inserting INPUT and GOTO commands. The results would be

```
---------------------------------------------------------------
10 INPUT X                     - Requests value for X
20 LET Y = 10*X + 7            - Computes value of 10X+7 and
30 PRINT Y                       assigns to Y. Y is printed.
40 GOTO 10                     - Tells computer to start at
50 end                           line 10 again.
---------------------------------------------------------------
```

This now means any number can be put in at the INPUT line, and line 40 tells the computer to restart at line 10. The program, in fact will never reach line 50. To stop this program you must press the CTRL & RESET keys simultaneously.

A better program should have line 5 which alerts the person running the program to enter the value for X.

```
-------------------------
5 PRINT"ENTER VALUE FOR X"
10 INPUT X
20 LET Y = 10*X + 7
30 PRINT Y
40 GOTO 5
50 END
-------------------------
```

Now, line 40 tells the computer to restart the program at the "ENTER VALUE FOR X" line.

EXERCISE

1. Show the output of the following program.

```
10 HOME
20 PRINT"ENTER BASE & HEIGHT";
30 INPUT B,H
40 LET A=B*H
50 PRINT"BASE","HEIGHT","AREA"
60 PRINT B,H,A
70 GOTO 20
80 END
```

2. The following program contains at least one error. Find each error and write a correct program.

```
10 PRINT"ENTER VALUES A,B";
20 INPUT A;B
30 PRINT"A*B = ";A*B
40 END
```

3. Modify the following program so that a person running the program will know to enter values for A and B. Show the output of program.

```
10 .......
20 INPUT A
30 .......
40 INPUT B
50 PRINT"A+B = ";A+B
60 END
```

4. Fill in the appropriate INPUT statement in lines 20 and 40 of the following program so that the values of A and B can be keyed into the computer memory during execution of the program.

```
10 PRINT"ENTER VALUE FOR A";
20 ...............
30 PRINT"ENTER VALUE FOR B";
40 ...............
50 PRINT A^2;B^2
60 END
```

5. Describe (analyze) what various programs are doing.

1.
```
10 LET X = 25
20 PRINT X
30 END
```

2.
```
10 LET Y = 1[illegible]
20 LET C = 50
30 PRINT
40 PRINT C,Y,C+Y
50 END
```

3.
```
5 LET M = 5
7 LET M = M+2
10 PRINT M
15 GOTO 7
25 END
```

4.
```
10 LET X = 37
20 LET Z = X+4
30 PRINT X,Z
40 END
```

5.
```
10 LET A = 5 + 8 + 3 - 2
20 PRINT A
30 END
```

6.
```
10 LET Z = 10
20 PRINT Z
30 LET Z = Z + 3
40 PRINT Z
55 END
```

7. What happens in the above questions when you add
50 GOTO 10

8. What happens in question 6 when you add
50 GOTO 30

9.
```
10 LET C = 1
20 LET D = 3
25 LET F = C + D
30 PRINT C, D, E
40 LET C = C + 1
50 LET D = D + C
60 GOTO 25
70 END
```

10.
```
10 PRINT "TYPE A NUMBER"
20 INPUT N
30 PRINT N, 2*N, N*N
40 END
```

11.
```
10 PRINT "TYPE FIRST NUMBER"
15 INPUT M
20 PRINT "TYPE SECOND NUMBER"
25 INPUT N
30 LET S = M + N
40 PRINT "THEIR SUM IS ";S
50 PRINT
60 GOTO 10
70 END
```

12.
```
10 LET N = 13
20 PRINT N
30 LET N = N + 11
40 GOTO 20
50 END
```

13.
```
10 REM AVERAGE OF 3 NUMBERS
20 PRINT"TYPE FIRST NUMBER"
30 INPUT A
40 PRINT"TYPE SECOND NUMBER"
50 INPUT B
60 PRINT"TYPE THIRD NUMBER"
70 INPUT C
80 LET M = (A + B + C)/3
90 PRINT "AVERAGE IS "; M
100 GOTO 20
110 END
```

14.
```
10 REM FIND VALUE FOR Y=5X+7
20 PRINT "ENTER VALUE FOR X"
30 INPUT X
40 LET Y = 5*X + 7
50 PRINT X;",";Y
60 GOTO 20
70 END
```

LESSON 4

4.1 REVIEW OF LESSON 3

4.2 IF-THEN STATEMENT

The IF-THEN statement makes a comparison of two numbers. It tells the computer what to do based on the results of the comparison.

The usual form of the IF-THEN statement is as follows.

IF algebraic sentence THEN line number

If the algebraic sentence is true, then the computer proceeds to the line number following THEN. If the algebraic sentence is false the computer simply goes in normal sequence to the next line.

However, instead of a line number after THEN a command may be written. For example,

```
10 IF X < 0 THEN PRINT "NOTE: X IS NEGATIVE"
20 LET Y = 15/X
```

So if the value for X is less than zero the screen will print the message and proceed to line 20. If X is, say, 5, the program will continue to line 20 (without printing anything) and assign Y the value of 15 ÷ 5 which is 3. This program, then, would alert the user to negative values of X.

The algebraic sentence in the IF-THEN statement must use one the following symbols:

BASIC Symbol	Meaning
=	is equal to
<	is less than
<=	is less than or equal to
>	is greater than
>=	is greater than or equal to
<>	is not equal to

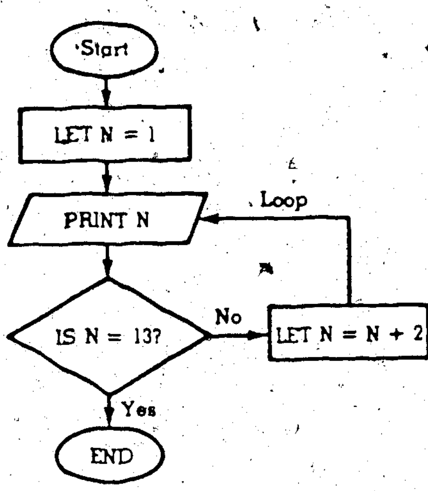
An Example of IF-THEN:

** Write a program which finds the square root of a number. If the number is negative, instruct the computer to print NO REAL SQUARE ROOT. Use INPUT statement.

```
10 PRINT"GIVE ME A NUMBER";
20 INPUT A
30 IF A<0 THEN 60                   -[Test line]
40 PRINT A^.5                       -[Print sq. root of A]
50 GOTO 10                          -[Why is this GOTO
60 PRINT "NO REAL SQUARE ROOT"        statement necessary?]
70 GOTO 10
80 END
```

4.3 FOR-NEXT LOOPS

Loops are often written in a program to direct the computer to repeat a portion of the program a certain number of times. Consider the following flow chart (For a description of the flowchart symbols see Flow Charting, page 34.) and program.




```
10 LET N=1
20 PRINT N;
30 IF N=13 THEN 60
40 LET N=N+2
50 GOTO 20
60 END
```

The same looping procedure can be accomplished by using FOR and NEXT.

```
10 FOR N=1 TO 13 STEP 2 ┐
20 PRINT N;             ├FOR-NEXT loop
30 NEXT N               ┘
40 END
```

A FOR-NEXT loop must begin with a FOR statement and end with a NEXT statement. The variable used in each statement must be the same. Any number of lines may appear between the FOR statement and the NEXT statement.

The general form of the FOR statement is as follows.

<u>FOR</u> Variable = Number <u>TO</u> Number <u>STEP</u> Number

If the step is not indicated, the computer will automatically use a step of one. The second number must be greater than the first number unless the step number is negative. The three numbers may also be variables that have had values assigned to them previously.

When the computer executes a FOR-NEXT loop, the computer automatically tests the value of the variable each time it is incremented. If it is less than or equal to the second number listed in the FOR statement, the computer increases the value by the indicated step and continues in the loop. Otherwise the computer goes to the first statement following the NEXT statement that can be executed.

An example of FOR-NEXT

** Write a program that computes Y values for the equation $Y = 8x^2 - 30x + 25$ when x = 1, 1.25, 1.50, 1.75, . . ., 3.

```
10 PRINT"X","Y"
20 FOR X=1 TO 3 STEP .25
30 LET Y=8*X^2-30*X+25
40 PRINT X,Y
50 NEXT X
60 END
```

More than one FOR-NEXT loop may occur in a program. There are only two ways that they can appear in a program.

Nested Loops	Independent Loops	Not Acceptable
┌FOR X │┌FOR Y │└NEXT Y └NEXT X	┌FOR X └NEXT X ┌FOR Y └NEXT Y	┌FOR X ├FOR Y │└NEXT X └NEXT Y
The loops do not cross.	The loops do not cross. They are not nested.	These loops cross.

EXERCISE

1. Suppose X=3, Y=10, and Z=15. State what line number the computer will go to after it executes the IF-THEN statement.

(a)
```
30 IF X>2 THEN 50
40 PRINT X
```

(b)
```
40 IF Y<>0 THEN 90
50 PRINT Y
```

(c)
```
30 IF X+Y<Z THEN 70
40 PRINT X^2+Y^2
```

(d)
```
50 IF Y^2>Z^2 THEN 70
60 PRINT Y
```

(e)
```
20 IF Y>=12 THEN 40
30 PRINT X*Y
```

(f)
```
20 IF X>Y THEN 50
30 IF Y<Z THEN 70
40 PRINT X+Z
```

2. Use the partial program at the right to tell whether A, B, or both A and B will be printed. Give the value of any variables printed.

(a) A = 19, B = 5

(b) A = 8, B = 8

(c) A = 13, B = 18

```
10 IF A>B THEN 50
20 LET A=A+10
30 LET B=B+2
40 IF A>=B THEN 60
50 PRINT A
60 PRINT B
70 END
```

3. Write the output of each of the following programs.

(a)
```
10 LET A=1
20 PRINT A,A^3
30 LET A=A+2
40 IF A<7 THEN 20
50 END
```

(b)
```
10 LET X=5
20 PRINT X
30 LET X=X-2
40 IF X<>-1 THEN 20
50 END
```

```
(c) 10 LET S=0
    20 LET N=1
    30 LET S=S+N
    40 IF N=4 THEN 70
    50 LET N=N+1
    60 GOTO 30
    70 PRINT S
    80 END
```

4. Write a FOR statement that assigns the numbers in each of the following lists to the variable X.

(a) 1, 2, 3, 4, 5, 6, 7, 8   (b) 0, 2, 4, 6, 8

(c) 2.25, 2.5, 2.75, . . . , 18   (d) -5, -3.5, -2, . . . , 9

5. Find the error(s) in each of the following programs and write a correct program. Then show the output of each program.

```
(a) 10 FOR X=1 TO -10
    20 NEXT X
    30 PRINT X
    40 END
```

```
(b) 10 FOR M=3 TO 25,STEP [illegible]
    20 PRINT M
    30 NEXT M
    40 PRINT M^2;M^.5
    50 END
```

```
(c) 10 LET X=3
    20 FOR C=1 TO 5
    30 PRINT C+X
    40 NEXT X
    50 END
```

```
(d) 10 FOR X=1 TO [illegible] STEP 3
    20 FOR Y=2 TO 4 STEP 2
    30 PRINT X*Y
    40 NEXT X
    50 NEXT Y
    60 END
```

6. Please do the following.

(a) Write a program to print your name 10 times.

(b) Write a program to request a name and print that name 10 times.

LESSON 5

5.1 REVIEW OF LESSON 4

5.2 SYSTEM COMMANDS TO COMMUNICATE TO THE DISK

1). INIT: The INIT command is used to prepare a disk in order that programs can be stored on it. When you purchase a new disk for the purpose of saving programs or data, it is completely blank and therefore will do nothing. Before you can store programs on a disk, it must first be "INITialized". If you want to clear a disk that has programs on it, it must also be "INITialized".

CAUTION! When a disk is INITialized everything stored on it is erased. Make sure you do not INITialize a disk that contains data you want to save.

The INIT command stores on the disk any program that is in memory when you use it. This becomes the "greeting program" which is automatically run every time you put the disk in the disk drive and turn on the power. The "greeting program" can be as simple or complex as you desire. The file name of the greeting program must be specified when you use the INIT command. It is your responsibility to make sure that there is always a program by that name on the disk. If you delete the greeting program, you will see the error message FILE NOT FOUND every time you turn the machine on. The only way to stop that error message is to put a program on that disk with the greeting's file name. That might be difficult to do if you do not know, or cannot remember, what file name was assigned, because there is no way of determining what the name of the greeting program is.

The best solution is prevention. Always specify the same greeting program file name when you initialize your disks. The standard greeting program name is HELLO.

To initialize a new disk first put the System Master Disk in the disk drive then turn the power on, remove it from the drive, and replace it with a new blank disk (or a used disk). Use the NEW command to clear the memory, then type in a greeting program. It is a good idea to test run the greeting program before it is stored on the disk to ensure that it does what you want it to do. Now enter the command: INIT HELLO. Make sure that the drive door is shut and press the RETURN key.

An Example of INIT HELLO:

```
------------------------------------------------------------
NEW
10 HOME
20 PRINT"MICHAEL JACKSON..........FEBRUARY 2/84"
30 PRINT"MATH 20-PROGRAMMING EXERCISES"
40 END
------------------------------------------------------------
```

2). <u>CATALOG</u>: To see what programs are on the disk, use the CATALOG command. Simply type CATALOG, then press RETURN key and a list of file names will appear on the screen.

3). <u>SAVE</u>: The SAVE command stores programs onto a disk. To store a program you have in computer memory, type in SAVE, a file name followed by a comma and disk drive number then press RETURN key. The file name may consist of up to 30 characters. Every file name must begin with a letter. The disk should whir and clack as the file is being written onto the disk. When SAVE is finished, the BASIC prompt and cursor will reappear on the screen. Type CATALOG to see if your program is listed on the disk directory.

An Example of SAVE:

SAVE SQUARE,D1 (hit RETURN key)

Since you must type a file name every time you wish to run the program, it is to your advantage to keep it as short as possible. Make sure, however, that the name relates to the program so that at a later date you will have no difficulty remembering its contents.

4). LOAD: The LOAD command loads programs into computer memory. This command does not run the program. To load a program into computer memory, type in LOAD, a file name followed by a comma and disk drive number then press RETURN key.

An Example of LOAD:

LOAD SQUARE,D1 (hit RETURN key)

---

5). DELETE: The DELETE command erases files from the disk. Use this command only when you are certain you will no longer require that particular program. To erase a file from your disk, type the DELETE command followed by the file name of the program you wish to erase from your disk.

An Example of DELETE:

DELETE SQUARE (hit RETURN key)

---

6). RENAME: The RENAME command changes the name of any file on the disk. To use the RENAME command, type in RENAME, the name of the old file followed by a comma and the new name you wish to give the file. Make sure that the new name you are giving the file has not already been used on that disk.

An Example of RENAME:

```
RENAME SQUARE,DOUBLE (hit RETURN key)
```

This will change the program named SQUARE to DOUBLE.

5.3 USING A PRINTER

To have your program print on the printer.

Load your program into computer memory.

Type PR#1 (hit RETURN key).

Then hold down CTRL key and press I

Type 80N (hit RETURN key)

Ignore the syntax error.

Then type LIST (hit RETURN key).

EXERCISE

1. State the system commands required by your computer to:

   (a) Store a program onto a disk.

   (b) Display a list of file names on the screen.

   (c) Change a program name.

   (d) Remove a program from a disk.

2. If you want to store a program named "PROBLEM #1" on to your disk, what format of command you should type in?

3. To remove a program named "EXERCISE #2" from your disk, what format of command you should type in?

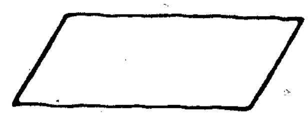
Flow Charting

A flow chart is a diagram that shows a step-by-step procedure for solving the problem. Its purpose is to clearly define an overall plan so that you can more readily visualize the logical flow of the program.

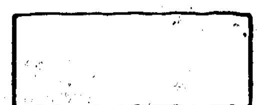
The shapes which are used to draw flow charts have special meanings.

An oval is used to begin or end a program.

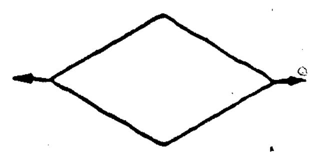
A parallelogram shows input or output. It is used with READ or PRINT statements.



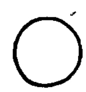
A rectangle shows processing operations. It is used with LET statements.



A diamond shows a decision. Arrows show how the flow continues. It is used with IF-THEN statements.

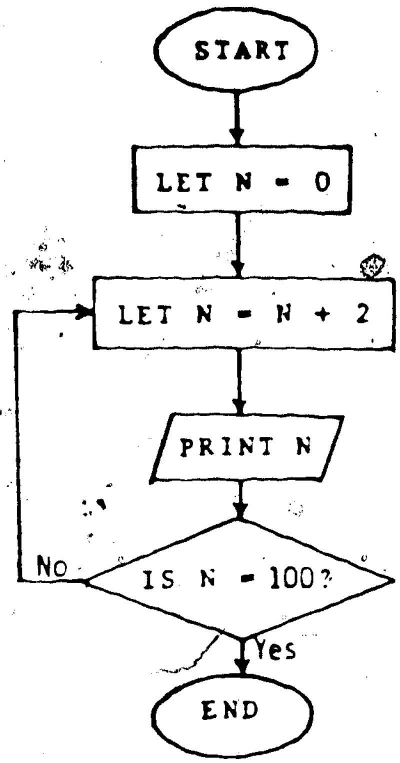


A circle is used to connect shapes when drawing an arrow is inconvenient. With an appropriate letter or number inside, one circle denotes an exit from one part of the flow chart, while a second circle indicates the entry to



another part of the flow chart, thus maintaining a continuous flow of logic.

Consider the following flow chart of the program on the right which prints the even numbers from 2 to 100 inclusive and stops at number 100.




```
10 LET N=0
20 LET N=N+2
30 PRINT N
40 IF N=100 THEN 60
50 GOTO 20
60 END
RUN
2
4
6
.
.
.
100
```

Answer Key for the Student Manual

Exercise 1

1. a) NEW   b) HOME

2. a) Press RETURN key

   b) Move cursor to that position and correct the character

   c) Press the Left Arrow key

3. Addition   +
   Subtraction   -
   Multiplication   *
   Division   /
   Raising to a power   ^

4. a) 5*X   b) 184/2   c) 2*(X+Y)   d) 3.5*Y   e) 2+X^2+3*X

5. a) Deferred Mode   b) Immediate Mode

6. EXERCISE #1 [illegible]

7. a) 7.[illegible]6e+0[illegible]   b) 1.27*123e-03   c) 5e-10

Exercise 2

1. a) LIST   b) RUN   c) DEL   d) END

2. a)
```
15 PRINT 6*(5+4)
20 END
```
   b)
```
10 PRINT 3^2
20 END
RUN
```
   c)
```
10 LET A=6
20 LET Y=A*3
30 PRINT Y
40 END
```

3. a) Yes   b) Yes   c) Yes   d) No   e) No   f) No

4. a) Yes   b) No   c) Yes   d) No   e) Yes   f) No

5. a) 15   b) 4   16
            7   49
            24

6. a) Add line number   b) At line 30 PRINT X*Y

   c) Variable A must be given value before line 20

Exercise 3

1. Example output.

```
ENTER BASE & HEIGHT? 2,4
BASE   HEIGHT   AREA
 2      4        8
```

2. At line 20, change A;B to A,B

3.
```
10 PRINT"ENTER A VALUE FOR A";
30 PRINT"ENTER A VALUE FOR B";
```

4.
```
20 INPUT A
40 INPUT B
```

Exercise 4

1. a) line 50   b) line 90   c) line 70

   d) line 60   e) line 30   f) line 70

2. a) Both values of A and B will be printed, 19 and 5

   b) Only value of B will be printed, 10

   c) Only value of B will be printed, 20

3. a) 1   1   b) 5   c) 10
      3   27   3
      5   125   1

4. a) FOR X=1 TO 8   b) FOR X=0 TO 8 STEP 2

   c) FOR X=2.25 TO 18 STEP 2.25   d) FOR X=-5 TO 9 STEP 1.5

5\. a) 10 FOR X=-10 TO 1 b) At line 10, delete , in front of STEP

Output: 1 Output: 3

5

7

.

.

.

25

729 5.19615243

c) At line 40 NEXT C d) At line 40 NEXT Y

50 NEXT X

Output: 4

5

6

7

8

Output: 2

4

8

16

6\. a)

```
10 FOR X=1 TO 10
20 PRINT"(YOUR NAME)"
30 NEXT X
40 END
```

b)

```
10 PRINT"WHAT IS YOUR NAME?"
20 INPUT X$
30 FOR X=1 TO 10
40 PRINT X$
50 NEXT X
60 END
```

Exercise 5

1\. a) SAVE b) CATALOG c) RENAME d) DELETE

2\. SAVE PROBLEM #1

3\. DELETE EXERCISE #2

APPENDIX II

STUDENT EXERCISES FOR PROGRAMMING IN BASIC - MATH 20 WITH SOLUTION PROGRAMS

Topic 1

Polynomials

1. Evaluate $g(x) = 5x^2 - 3x + 2$ at $g(3)$, $g(-2)$, $g(0)$.
2. If $h(x) = 2x^2 - 4x + 3$, evaluate $h(2)-h(-3)+h(1)$.
3. Find the zero of the following linear functions

   a) $y = 3x - 6$

   b) $y = ax + b$
4. Calculate approximate solutions for the following equations by trying different values for x. First use integer to find where the value of the polynomial changes sign and then narrow down this range.

   a) $x^2 - 6x + 3 = 0$

   b) $x^2 - x - 6 = 0$

   c) $ax^2 + bx + c = 0$
5. Given $P(x) = x^2 + 6x + 5$, determine if the polynomial has zeros for x values in the domain $-10 \leq x \leq 10$.
6. Make a table of ordered pairs of $y = 5x^2 - 3x + 2$ for values of x from -5 to 5 inclusive.
7. Write a program to

   a) derive a table of values using GOTO

   b) derive ordered pairs using FOR-NEXT

   c) derive ordered pairs using STEP

   d) derive ordered pairs for any quadratic function of the form

   $y = ax^2 + bx + c$
8. Factor $x^2 + 5x + 6$

Topic 2

Radicals

1. Evaluate $(\sqrt{2})^4$, $(\sqrt{3})^5$.
2. Write the following as single numbers.

   a) $\sqrt{16/25}$ b) $-\sqrt{144/9}$

   c) $\sqrt[3]{-125/64}$ d) $\sqrt[4]{81/625}$
3. Evaluate y = 2 x for integral values of x from 1 to 10.
4. Given any mixed radical, write it as a complete radical.

   eg. $5\sqrt{2}$ = , $2\sqrt{3}$ = ?, $4\sqrt{5}$ = ?
5. Evaluate a radical of the form

   a) $\sqrt[n]{R}$ b) $\sqrt[n]{R^m}$ or $(\sqrt[n]{R})^m$

Topic 3

Relations and Functions

1. For the function $y = 3x + 2$, $1 \leqslant x \leqslant 12$, $x \in N$ state the range.
2. For the function $y = x^2$, $-4 \leqslant x \leqslant 4$, $x \in I$ state the range.
3. List the y values for the equation $y = 3x^2 + 2$ given $-5 \leqslant x \leqslant 7$, $x \in$
4. List the ordered pairs for $y = -3x + 7$, where $-2.0 \leqslant x \leqslant -0.8$, x changing in steps of 0.2. Print the equation at the top and then list the ordered pairs in 2 columns titled

   X VALUES   Y VALUES
5. Evaluate y in the equation $y = x^2 - 7x + 2$ for any x the user wants to input. Write the program so that the user will be given suitable instructions, and output that will be understandable.

Topic 4

Coordinate Geometry

The Straight Line

1. Find the slope of $3y - 2x + 3 = 0$.
2. Find the slope of the general linear function $Ax + By + C = 0$ and use it to determine the slope of any linear function in standard form ($Ax + By + C = 0$).
3. Given the two points (6,1) and (-2,3) determine the equation of the line passing through these points.
4. Find the x- and y-intercepts of $y = 3x + 7$.
5. Determine if the two equations are parallel.

   $y + 6x - 15 = 0$, $2x = 5 + y$.
6. Find the x- and y-intercepts of an equation of the form $Ax + By + C = 0$. If the equation does not have a particular intercept, a message should be displayed on the screen telling the user why. The user is to input the values of A, B, and C.
7. For a pair of straight lines described by the equations

   $Ax + By + C = 0$

   $Dx + Ey + F = 0$

   determine whether the lines are - parallel

   - perpendicular
   - neither of these.

   The user is to input the values of A, B, C, D, E, and F.
8. Write a program which will list suitable pairs which will enable you to sketch a graph of

   a) $y = 3x + 6$

   b) $y = mx + b$

Topic 5

Coordinate Geometry

1. Find the distance between (1,2) and (5,-2).
2. Find the distance between the point (A,B) and the point (C,D). The user is to input the values of A, B, C, and D.
3. Find the coordinates of the midpoint between the points (A,B) and (C,D). The user is to input A, B, C, and D.
4. Write a program for the slope between the points (-3,5) & (7,2).
5. Write a program to determine if the following three points are colinear: (1,2), (2,4), and (5,3).

Topic 6

System of Equations

1. Solve the following systems of equations by the method of substitution.

   $4x - 3y = 6, \quad y = 3$

2. Determine whether the following systems are independent, dependent, inconsistent.

   a) $y = 3x + 5, \; 2y = 6x + 5$

   b) $2x + y = -1, \; -4x + y = 11$

   c) $y = 2x + 1, \; y = x$

   d) $y = 3x + 5, \; 2y = 6x + 10$

3. Given two equations of the form

   $Ax + By + C = 0$

   $Dx + Ey + F = 0$

   determine if the system has a unique solution. The user is to input the values of A, B, C, D, E, and F.

Topic 7

Variation

1. How far apart are two people after 6 hours if they travel in the same direction at 10 km/hr and 15 km/hr?
2. The formula for blood pressure according to age is $P = 100+(1/2)A$, where A is age. Make a chart in 5 year intervals from 10 to 80 years.
3. Solving any problem involving direct variation when the general equation is in the form $y = cx$, where c is the constant.
4. Solving any problem involving inverse variation when the general equation is in the form $y = c/x$.

Topic 8

Geometry

The Circle

1. A point P is 25 cm. from the centre of a circle of radius 7 cm. Find the distance from point P to the point of tangency.
2. A circle has radius 5 cm. A chord is 6 cm., long. How far is the chord from the centre?
3. Given a central angle, determine the measure of the inscribed angle. Give the user suitable instructions and an understandable answer.
4. Finding the length of an arc given the measure of the sector angle and the radius.
5. Finding the area of a sector given the measure of the sector angle and the radius.

Topic 9

Trigonometry

1. Is a triangle of sides 3 cm, 10 cm, and 11.32 cm a right angle triangle?
2. Given Δ ABC with A = 30°, B = 60°, and C = 90° and a = 1, b = 3, c = 2, find sin 30°, cos 30°, tan 30°, sin 60°, cos 60°, and tan 60°.
3. What is the value of sin θ if (3,8) lies on the terminal side of angle θ.
4. If cosine θ = 0.7 find values of sin θ and tan θ. Assume θ is between 0° and 90°.
5. Given an angle θ in degrees, write a program to convert it to radians.
6. Write a program that will display sin θ, cos θ, tan θ if the angle is input in degrees.

Topic 10

Statistics

1. Place the following numbers in groups of width 1000.

   500, 2503, 3682, 1032, 5296, 10362, 9827, 8365, 5021, 5932, 6289, 8921, 7660, 4032.

   Find the frequency in each category.
2. Find the mean of any given series of numbers.
3. Find the median of any given set of 5 numbers. The user is to input the numbers in ascending or descending order, ie. sorting.
4. Determine Joe's percentile if on a test he was ranked 18th out of 60.

Topic 11

Quadratic Functions

1. Find the axis of symmetry for $y = 4x^2 + 4x + 5$.
2. Find the vertex, axis of symmetry, and the point of intersection with the y-axis of $y = 4x^2 + 4x + 5$.
3. Find the maximum or minimum value of $y = -3x^2 - 6x + 5$.
4. Given an equation of the form $Ax^2 + Bx + C = 0$, determine its roots. Check, and display an appropriate message on the screen telling the user whether the equation has

   - no real roots
   - one real root
   - two real roots.

   The user is to input A, B, and C.
5. Determine the range of $y = 4x^2 + 4x + 5$
6. Write a program to list suitable pairs which will enable you to sketch the graph of

   a) $y = x^2$

   b) $y = 5x^2$

   c) $y = (1/3)x^2$

   d) $y = 2x^2 + 3$

   e) $y = ax^2 + bx + c$
7. Evaluate the discriminant for

   a) $y = x^2 - x - 12$

   b) $y = 4x^2 - 4x + 1$

   c) $y = ax^2 + bx + c$

   (Have the computers print the characteristics of the roots. Real, non-real, etc.)

Topic 1

Polynomials

Problem 1

```
10 REM EVALUATE G(X) = 5X^2-3X+2
20 PRINT"ENTER A VALUE FOR X";
30 INPUT X
40 LET Y=5*(X^2)-(3*X)+2
50 PRINT"X = "X", G("X") = "Y
60 GOTO 10
70 END
```

Problem 2

```
10 REM EVALUATE H(X) = 2X^2-4X+3
20 FOR I=1 TO 3
30 PRINT"ENTER #"I" VALUE OF X";
40 INPUT X(I)
50 LET H(I)=2*(X(I)^2)-(4*X(I))+3
60 NEXT I
70 FOR I=1 TO 3
80 PRINT"H("X(I)") = "H(I)
90 NEXT I
100 LET Y=H(1)-H(2)+H(3)
110 PRINT"H("X(1)")-H("X(2)")+H("X(3)") = "Y
120 END
```

Problem 3

```
10 REM FIND THE ZERO OF FUNCTION Y = AX+B
20 PRINT"ENTER VALUES FOR A, B";
30 INPUT A,B
40 LET S=-B/A
50 PRINT"Y = 0 WHEN X = "S
60 GOTO 10
70 END
```

Problem 4

```
10 REM APPROXIMATE SOLUTIONS FOR A(X^2)+BX+C = 0
20 INPUT"ENTER VALUES FOR A, B, C";A,B,C
30 PRINT"Y = "A"(X^2)+"B"X+"C
40 INPUT"ENTER A VALUE FOR X";X
50 LET Y=A*(X^2)+B*X+C
60 PRINT"X = "X": Y = "Y
70 GOTO 40
80 END
```

Problem 5

```
10 PRINT"P(X) = X^2+6X+5"
20 FOR X=-10 TO 10
30 LET P=(X^2)+(6*X)+5
40 IF P=0 THEN PRINT"X = "X,"P = "P
50 NEXT X
60 END
```

Problem 6

```
10 PRINT"Y = 5(X^2)-3X+2, -5<=X<=5"
20 PRINT"(X , Y)"
30 FOR X=-5 TO 5
40 LET Y=5*(X^2)-(3*X)+2
50 PRINT"("X","Y")"
60 NEXT X
70 END
```

Problem 8

```
10 PRINT"FACTOR POLYNOMIAL OF THE FORM (X^2)+BX+C"
20 INPUT"ENTER VALUES FOR B,C";B,C
30 IF C<0 THEN 70
40 IF B<0 THEN 60
50 FOR M=-B TO B:GOTO 80
60 FOR M=B TO -B:GOTO 80
70 FOR M=C TO -C
80 IF M*(B-M)<>C THEN 100
90 PRINT"X^2+"B"X+"C" = (X+"M")(X+"B-M")"
100 NEXT M:GOTO 10
110 END
```

Solution Programs

Topic 2

Radicals

Problem 1

```
10 PRINT"EVALUATE RADICAL OF THE FORM (A^(1/N))^C"
20 INPUT"ENTER A VALUE OF RADICAND";A
30 INPUT"ENTER AN INDEX OF RADICAL";N
40 INPUT"ENTER THE EXPONENT NUMBER";C
50 LET Y=(A^(1/N))^C
60 PRINT"("A"^(1/"N"))^"C" = "Y
70 GOTO 10
80 END
```

Problem 2

```
10 PRINT"EVALUATE RADICAL OF THE FORM A^(1/N)"
20 INPUT"ENTER A VALUE OF RADICAND";A
30 INPUT"ENTER AN INDEX OF RADICAL";N
40 LET Y=A^(1/N)
50 PRINT A"^(1/"N") = "Y
60 GOTO 10
70 END
```

Problem 3

```
10 PRINT"EVALUATE RADICAL OF THE FORM Y = 2(X^.5)"
20 FOR X=1 TO 10
30 LET Y=2*(X^.5)
40 PRINT"X = "X,"Y = "Y
50 NEXT X
60 END
```

Problem 4

```
10 REM WRITE ANY MIXED RADICAL AS A COMPLETE RADICAL
20 INPUT"ENTER A POSITIVE REAL NUMBER";A
30 INPUT"ENTER A VALUE OF RADICAND";B
40 INPUT"ENTER AN INDEX OF RADICAL";N
50 LET Y=(A^N)*B
60 PRINT A"*("B"^(1/"N")) = "Y"^(1/"N")"
70 GOTO 10
80 END
```

Topic 3

Relations and Functions

Problem 1

```
1Ø PRINT"EVALUATE Y=3X+2, 1<=X<=12"
2Ø FOR X=1 TO 12
3Ø LET Y=(3*X)+2
4Ø PRINT"X = "X,"Y = "Y
5Ø NEXT X
6Ø END
```

Problem 2

```
1Ø PRINT"EVALUATE Y=X 2, -4<=X<=4"
2Ø FOR X=-4 TO 4
3Ø LET Y=X^2
4Ø PRINT"X = "X,"Y = "Y
5Ø NEXT X
6Ø END
```

Problem 3

```
1Ø PRINT"LIST THE Y VALUES OF Y=3(X^2)+2, -5<=X<=7"
2Ø FOR X=-5 TO 7
3Ø LET Y=3*(X^2)+2
4Ø PRINT"Y = "Y
5Ø NEXT X
6Ø END
```

Problem 4

```
1Ø PRINT"LIST THE ORDERED PAIRS FOR Y=-3X+7, -2.Ø<=X<=-Ø.8
2Ø PRINT"X VALUES","Y VALUES"
3Ø FOR X=-2.Ø TO -Ø.8 STEP Ø.2
4Ø LET Y=(-3*X)+7
5Ø PRINT X,Y
6Ø NEXT X
7Ø END
```

Problem 5

```
1Ø PRINT"ENTER A VALUE OF X FOR Y=(X^2)-7X+2"
2Ø INPUT X
3Ø LET Y=(X^2)-(7*X)+2
4Ø PRINT"THE ANSWER IS Y = "Y
5Ø GOTO 1Ø
6Ø END
```

Topic 4

Coordinate Geometry

(The Straight Line)

Problem 1

```
10 REM FIND SLOPE OF 3Y-2X+3 = Ø
2Ø LET A=-2
3Ø LET B=3
4Ø PRINT"SLOPE = ";-A;"/";B
5Ø END
```

Problem 2

```
1Ø REM FIND SLOPE OF AX+BY+C = Ø
2Ø INPUT"ENTER VALUES FOR A, B, C";A,B,C
3Ø PRINT"LINEAR FUNCTION: "A"X+"B"Y+"C" = Ø"
4Ø IF B=Ø THEN PRINT"SLOPE IS UNDEFINED.":GOTO 2Ø
5Ø PRINT"SLOPE = "(-1)*A"/"B
6Ø GOTO 2Ø
7Ø END
```

Problem 3

```
1Ø REM TWO POINTS EQUATION
2Ø LET X1=6:Y1=1
3Ø LET X2=-2:Y2=3
4Ø PRINT"THE LINE IS Y-"Y1" = "(Y2-Y1)"/"(X2-X1)"(X-"X1")"
5Ø END
```

Problem 4

```
1Ø REM FIND X- AND Y-INTERCEPT OF Y=3X+7
2Ø LET A=3
3Ø LET C=7
4Ø PRINT"X-INTERCEPT IS ("(-1*C)/A",Ø)"
5Ø PRINT"Y-INTERCEPT IS (Ø,"C")"
6Ø END
```

Problem 5

```
1Ø PRINT"DETERMINE IF Y+6X-15=Ø, 2X=5+Y ARE PARALLEL"
2Ø LET A=6:B=1
3Ø LET D=2:E=-1
4Ø LET M1=-A/B
5Ø LET M2=-D/E
6Ø PRINT"M1 = "M1,"M2 = "M2
7Ø IF M1=M2 THEN PRINT"LINES ARE PARALLEL":END
8Ø PRINT"LINES ARE NOT PARALLEL"
9Ø END
```

Problem 6

```
10 PRINT"FIND X- AND Y-INTERCEPTS OF AX+BY+C = Ø"
20 INPUT"ENTER VALUES FOR A, B, C";A,B,C
30 IF A=Ø THEN PRINT"NO X-INTERCEPT. LINE // TO X-AXIS":
                                                  GOTO 1Ø
40 IF B=Ø THEN PRINT"NO Y-INTERCEPT. LINE // TO Y-AXIS":
                                                  GOTO 1Ø
50 LET M=-C/A
60 LET N=-C/B
70 PRINT"X-INTERCEPT IS ("M",Ø)"
80 PRINT"Y-INTERCEPT IS (Ø,"N")"
90 GOTO 1Ø
1ØØ END
```

Problem 7

```
1Ø PRINT"DETERMINE AX+BY+C=Ø & DX+EY+F=Ø ARE PARALLEL"
2Ø PRINT"OR PERPENDICULAR"
3Ø INPUT"ENTER VALUES FOR A, B, C";A,B,C
4Ø INPUT"ENTER VALUES FOR D, E, F";D,E,F
5Ø IF B=Ø AND E=Ø THEN PRINT"SLOPES OF BOTH LINES ARE
                                    UNDEFINED":GOTO 1Ø
6Ø IF B=Ø THEN PRINT"SLOPE OF 1ST LINE IS UNDEFINED":GOTO 1Ø
7Ø IF E=Ø THEN PRINT"SLOPE OF 2ND LINE IS UNDEFINED":GOTO 1Ø
8Ø LET M1=-A/B: M2=-D/E
9Ø IF M1=M2 THEN PRINT"LINES ARE PARALLEL":GOTO 1Ø
1ØØ IF M1*M2=-1 THEN PRINT"LINES ARE PERPENDICULAR":GOTO 1Ø
11Ø PRINT"LINES ARE NEITHER PARALLEL NOR PERPENDICULAR":
                                                 GOTO 1Ø
12Ø END
```

Topic 5

Coordinate Geometry

Problem 1

```
10 PRINT"FIND THE DISTANCE BETWEEN (1,2) AND (5,-2)"
20 LET X1=1:Y1=2
30 LET X2=5:y2=-2
40 LET D=((X2-X1)^2+(Y2-Y1)^2)^.5
50 PRINT"THE DISTANCE = "D
60 END
```

Problem 2

```
10 PRINT"FIND DISTANCE BETWEEN POINTS (A,B) AND (C,D)"
20 INPUT"ENTER A VALUE FOR A";A
30 INPUT"ENTER A VALUE FOR B";B
40 INPUT"ENTER A VALUE FOR C";C
50 INPUT"ENTER A VALUE FOR D";D
60 LET X=(C-D)^2
70 LET Y=(D-B)^2
80 LET Z=(X+Y)^.5
90 PRINT"THE DISTANCE BETWEEN THE POINTS = ";Z
100 GOTO 10
110 END
```

Problem 3

```
10 PRINT"FIND THE MIDPOINT OF (A,B) AND (C,D)"
20 INPUT"WHAT IS THE VALUE OF A";A
30 INPUT"WHAT IS THE VALUE OF B";B
40 INPUT"WHAT IS THE VALUE OF C";C
50 INPUT"WHAT IS THE VALUE OF D";D
60 LET X=(A+C)/2
70 LET Y=(B+D)/2
80 PRINT"THE MIDPOINT IS ("X","Y")"
90 GOTO 10
100 END
```

Topic 6

System of Equations

Problem 1

```
10 REM * SOLVE SYSTEM OF EQUATIONS BY SUBSTITUTION
20 LET Y=3
30 LET X=(3*Y+6)/4
40 PRINT"4X-3Y = 6, Y = 3"
50 PRINT"X = "X
60 END
```

Problem 2

```
10 REM DETERMINE WHETHER LINEAR SYSTEM OF THE FORM AX+BY=C
20 REM AND PX+QY=R ARE CONSISTENT, INCONSISTENT, DEPENDENT
30 REM OR INDEPENDENT
40 INPUT"ENTER VALUES FOR A, B, C";A,B,C
50 INPUT"ENTER VALUES FOR P, Q, R";P,Q,R
60 IF A<>0 AND B<>0 AND C<>0 AND P<>0 AND Q<>0 AND R<>0
                                                   THEN 160
70 IF A<>0 AND B<>0 AND C=0 AND P<>0 AND Q<>0 AND R=0
                                                   THEN 190
80 IF A<>0 AND B<>0 AND C=0 AND P<>0 AND Q<>0 AND R<>0
                                                   THEN 170
90 IF A<>0 AND B<>0 AND C<>0 AND P<>0 AND Q<>0 AND R=0
                                                   THEN 200
100 IF A<>0 AND B=0 AND C<>0 AND P<>0 AND Q=0 AND R<>0
                                                   THEN 220
110 IF A=0 AND B<>0 AND C<>0 AND P=0 AND Q<>0 AND R<>0
                                                   THEN 240
120 IF A<>0 AND B=0 AND C=0 AND P<>0 AND Q=0 AND R=0 THEN 260
130 IF A=0 AND B<>0 AND C=0 AND P=0 AND Q<>0 AND R=0 THEN 260
140 IF A<>0 AND B=0 AND C<>0 OR C=0 AND P=0 AND Q<>0 AND
                                         R<>0 OR R=0 THEN 280
150 IF A=0 AND B<>0 AND C<>0 OR C=0 AND P<>0 AND Q=0 AND
                                         R<>0 OR R=0 THEN 280
160 IF A/P=B/Q AND B/Q=C/R THEN 260
170 IF A/P=B/Q AND B/Q<>C/R THEN 270
180 IF A/P<>B/Q THEN 280
190 IF A/P=B/Q THEN 260
200 IF P/A=Q/B AND Q/B<>R/C AND R/C=0 THEN 270
210 IF P/A<>Q/B THEN 280
220 IF A/P=C/R THEN 260
230 IF A/P<>C/R THEN 270
240 IF B/Q=C/R THEN 260
250 IF B/Q<>C/R THEN 270
260 PRINT"THE SYSTEM IS CONSISTENT AND DEPENDENT":END
270 PRINT"THE SYSTEM IS INCONSISTENT":END
280 PRINT"THE SYSTEM IS CONSISTENT AND INDEPENDENT":END
```

Problem 3

```
10 PRINT"DETERMINE IF AX+BY+C = 0 AND DX+EY+F = 0"
20 PRINT"HAVE A UNIQUE SOLUTION"
30 INPUT"ENTER VALUES FOR A, B, C";A,B,C
40 INPUT"ENTER VALUES FOR D, E, F";D,E,F
50 IF A/D=B/E AND B/E<>C/F THEN PRINT"THE SYSTEM HAS NO
                                        SOLUTIONS":GOTO 10
60 IF A/D<>B/E THEN PRINT"THE SYSTEM HAS A UNIQUE
                                        SOLUTION":GOTO 10
70 PRINT"THE SYSTEM HAS AN INFINITE NUMBERS OF SOLUTIONS":
                                                GOTO 10
80 END
```

Topic 7

Variations

Problem 1

```
10 REM DISTANCE BETWEEN TWO PEOPLE AFTER 6 HOURS
20 LET D1=6*10
30 LET D2=6*15
40 PRINT"AFTER 6 HRS PERSON A TRAVELED "D1" KMS."
50 PRINT"AFTER 6 HRS PERSON B TRAVELED "D2" KMS."
60 PRINT"DISTANCE BETWEEN 2 PERSONS IS "D2-D1" KMS."
70 END
```

Problem 2

```
10 PRINT"BLOOD PRESSURE FORMULA  IS P = 100+(1/2)A"
20 PRINT"   AGE","CLASS AGE","BLOOD PRESSURE"
30 FOR A=10 TO 75 STEP 5
40 LET P=100+(1/2)*(2*A+5)/2
50 PRINT A"-"A+5,(2*A+5)/2,P
60 NEXT A
70 END
```

Problem 3

```
10 PRINT"S^2 VARIES DIRECTLY AS T^3, S=2 WHEN T=4"
20 PRINT"FIND S WHEN T=8, FIND T WHEN S=16"
30 PRINT"S^2 = C(T^3) OR C = (S^2)/(T^3)"
40 LET S=2: T=4
50 LET C=(S^2)/(T^3)
60 PRINT"C = "C
70 LET T=8: S=(C*(T^3))^.5
80 PRINT"WHEN T=8, S="S
90 LET S=16: T=((S^2)/C)^(1/3)
100 PRINT"WHEN S=16, T="T
110 END
```

Problem 4

```
10 PRINT"X VARIES INVERSELY AS Y^2, X=3.5 WHEN Y=5"
20 PRINT"CALCULATE VALUE  OF Y WHEN X=14"
30 PRINT"X = C/(Y^2) OR C = X(Y^2)"
40 LET X=3.5: Y=5
50 LET C=X*(Y^2)
60 PRINT"C = "C
70 PRINT"Y = (C/X)^.5"
80 LET X=14
90 LET Y=(C/X)^.5
100 PRINT"WHEN X = 14, Y = "Y
110 END
```

Topic 8

Geometry

(The Circle)

Problem 1

```
10 PRINT"FIND DISTANCE FROM POINT P TO POINT OF TANGENCY"
20 LET A=25: B=7
30 LET C=(A^2-B^2)^.5
40 PRINT"DISTANCE FROM P TO POINT OF TANGENCY = "C" CM."
50 END
```

Problem 2

```
10 REM * DISTANCE FROM CHORD TO CENTRE OF A CIRCLE
20 LET A=5: B=3
30 LET C=(A^2-B^2)^.5
40 PRINT"DISTANCE FROM CHORD TO CENTRE OF CIRCLE = "C" CM"
50 END
```

Problem 3

```
10 PRINT"GIVE ME A CENTRAL ANGLE IN DEGREE";
20 INPUT C
30 LET B=C/2
40 PRINT"THE INSCRIBED ANGLE = "B" DEGREES"
50 GOTO 10
60 END
```

Problem 4

```
10 PRINT"GIVEN SECTOR ANGLE AND RADIUS, FIND ARCLENGTH"
20 PRINT"(USE PI = 3.14159)"
30 PRINT"IF SECTOR ANGLE IS IN DEGREES PRESS 1,"
40 PRINT"IN RADIAN PRESS 2";
50 INPUT X
60 IF X=1 THEN 90
70 INPUT"ENTER SECTOR ANGLE IN RADIAN";S
80 GOTO 110
90 INPUT"ENTER SECTOR ANGLE IN DEGREE";D
100 LET S=(D/180)*3.14159
110 INPUT"ENTER RADIUS OF THE CIRCLE";R
120 LET L=R*S
130 PRINT"THE ARCLENGTH = "L
140 GOTO 10
150 END
```

Problem 5

```
10 PRINT"GIVEN SECTOR ANGLE, FIND AREA OF SECTOR"
20 PRINT"(USE PI = 3.14159)"
30 PRINT"IF SECTOR ANGLE IS IN DEGREE PRESS 1,"
40 PRINT"IN RADIAN PRESS 2";
50 INPUT X
60 IF X=1 THEN 90
70 INPUT"ENTER SECTOR ANGLE IN RADIAN";S
80 GOTO 110
90 INPUT"ENTER SECTOR ANGLE IN DEGREE";D
100 LET S=(D/180)*3.14159
110 INPUT"ENTER RADIUS OF THE CIRCLE";R
120 LET A=.5*(R^2)*S
130 PRINT"AREA OF THE SECTOR = "A" SQUARE UNITS"
140 GOTO 10
150 END
```

Topic 9

Trigonometry

Problem 1

```
10 REM DETERMINE WHETHER A RIGHT ANGLE TRIANGLE
20 INPUT"ENTER VALUES FOR SIDE A, B, C";A,B,C
30 IF A^2+B^2=C^2 THEN 80
40 IF A^2+C^2=B^2 THEN 80
50 IF B^2+C^2=A^2 THEN 80
60 PRINTA" "B" "C" ARE NOT SIDES OF A RIGHT ANGLE TRIANGLE
70 GOTO 20
80 PRINT A" "B" "C" ARE SIDES OF A RIGHT ANGLE TRIANGLE"
90 GOTO 20
100 END
```

Problem 2

```
10 REM FIND SIN A, COS A, TAN A, A=30 AND 60 DEGREES
20 LET A=1: B=3^.5: C=2
30 PRINT"SIN 30 = "A/C
40 PRINT"COS 30 = "B/C
50 PRINT"TAN 30 = "A/B
60 PRINT"SIN 60 = "B/C
70 PRINT"COS 60 = "A/C
80 PRINT"TAN 60 = "B/A
90 END
```

Problem 3

```
10 REM FIND SIN A
20 LET X=3: Y=8
30 LET Z=(X^2+Y^2)^.5
40 PRINT"SIN A = "Y/Z
50 END
```

Problem 4

```
10 REM  COS A = 0.7, FIND SIN A AND TAN A
20 LET X=.7
30 LET Y=(1-X^2)^.5
40 PRINT"SIN A = "Y
50 PRINT"TAN A = "Y/X
60 END
```

Problem 5

```
10 REM CONVERT DEGREES TO RADIANS
20 PRINT"GIVE ME AN ANGLE IN DEGREE";
30 INPUT D
40 LET R=(D/180)*3.14159
50 PRINT D" DEGREES = "R" RADIANS"
60 GOTO 20
70 END
```

Problem 6

```
10 REM INPUT ANGLE (DEGREES), DISPLAY SIN A, COS A, TAN A
20 INPUT"WHAT IS YOUR DEGREE ANGLE";D
30 LET R=(D/180)*3.14159
40 PRINT"SIN "A" = "SIN(R)
50 PRINT"COS "A" = "COS(R)
60 PRINT"TAN "A" = "TAN(R)
70 GOTO 20
80 END
```

Topic 10

Statistics

Problem 1

```
10 REM THIS PROGRAM WILL PLACE NUMBERS IN CLASSES OF WIDTH
20 REM 1000 AND FIND FREQUENCY IN EACH CATEGORY AND YOU
30 REM CAN MODIFY THE PROGRAM TO FIND MEAN OF GROUPED DATA
40 REM YOU DON'T HAVE TO MAKE AN ARRAY OF THE DATA
50 INPUT"HOW MANY NUMBERS DO YOU HAVE";N
60 DIM X(N)
70 FOR I=1 TO N
80 PRINT"ENTER YOUR #";I;" NUMBER";
90 INPUT X(I)
100 NEXT I
110 INPUT"WHAT IS THE LOWEST NUMBER";A
120 INPUT"WHAT IS THE HIGHEST NUMBER";B
130 LET K=INT((B-A)/1000)+1
140 PRINT"NUMBER OF CLASSES = ";K;" (CLASS WIDTH = 1000)"
150 REM DELAY THE PROGRAM
160 FOR D=1 TO 1500:NEXT D
170 INPUT"ENTER A LOWER BOUNDARY OF THE LOWEST CLASS";M
180 LET C(J)=0
190 REM COUNT FREQUENCY FOR EACH CLASS
200 FOR I=1 TO N
210 FOR J=1 TO K
220 REM L=LOWER BOUNDARY, U=UPPER BOUNDARY
230 LET L=M+1000*(J-1): U=M+1000*(J)
240 IF L<=X(I) AND X(I)<U THEN C(J)=C(J)+1
250 NEXT J
260 NEXT I
270 PRINT"=========================="
280 PRINT"    CLASS      FREQUENCY"
290 PRINT"=========================="
300 FOR J=1 TO K
310 PRINT L;"-";U;"     ";C(J)
320 NEXT J
330 PRINT"=========================="
340 PRINT"           TOTAL   ";N
350 PRINT"=========================="
360 END
```

Problem 2

```
10 REM FIND MEAN OF ANY GIVEN SERIES OF NUMBERS
20 PRINT"HOW MANY NUMBERS DO YOU HAVE";N
30 INPUT N
40 LET Y=0
50 FOR I=1 TO N
60 PRINT"ENTER YOUR #"I" NUMBER";
70 INPUT X(I)
80 LET Y=Y+X(I)
90 NEXT I
100 PRINT"MEAN = "Y/N
110 GOTO 20
120 END
```

Problem 3

```
10 REM FIND MEDIAN OF ANY GIVEN 5 NUMBERS
20 PRINT"ENTER 5 NUMBERS IN ASCENDING OR DESCENDING ORDER"
30 FOR I=1 TO 5
40 PRINT"ENTER YOUR #"I" NUMBER";
50 INPUT X(I)
60 NEXT I
70 PRINT"THE MEDIAN IS "X(3)
80 GOTO 20
90 END
```

General Program for Problem 3

```
10 REM THIS PROGRAM FIND MEDIAN OF ANY GIVEN NUMBERS
20 REM WITHOUT SORTING THE NUMBERS
30 INPUT"HOW MANY NUMBERS DO YOU HAVE";N
40 FOR I=1 TO N
50 PRINT"ENTER YOUR #"I" NUMBER";
60 INPUT X(I)
70 NEXT I
80 REM THIS LOOP SORTING THE NUMBERS
90 FOR I=1 TO N-1
100 FOR J=I+1 TO N
110 IF X(I)<X(J) THEN 150
120 LET X=X(I)
130 LET X(I)=X(J)
140 LET X(J)=X
150 NEXT J
160 NEXT I
170 IF N/2=INT(N/2) THEN 200
180 LET M=INT(N/2)+1
190 PRINT"THE MEDIAN = "X(M): GOTO 30
200 LET M=X(N/2)+X(1+N/2)
210 PRINT"THE MEDIAN = "M/2: GOTO 30
220 END
```

Problem 4

```
10 REM DETERMINE PERCENTILE
20 INPUT"HOW MANY STUDENTS IN THE CLASS";N
30 INPUT"WHAT IS YOUR RANK";R
40 LET A=((R-1)*100)/N
50 PRINT"THE PERCENTILE IS "A
60 END
```

Topic 11

Quadratic Functions

Problem 1 & 2

```
10 REM FIND AXIS OF SYMMETRY, VERTEX, POINT OF
20 REM INTERSECTION WITH Y-AXIS OF Y=A(X^2)+BX+C
30 INPUT"ENTER VALUES FOR A, B, C";A,B,C
40 LET X=-B/(2*A)
50 LET Y=-(B*B-4*A*C)/(4*A)
60 PRINT"THE AXIS OF SYMMETRY IS X = "X
70 PRINT"THE POINT OF VERTEX IS ("X","Y")"
80 PRINT"GRAPH INTERSECTS Y-AXIS AT (0,"C")"
90 GOTO 30
100 END
```

Problem 3

```
10 REM FIND MAX. AND MIN. VALUE OF Y=A(X^2)+BX+C
20 INPUT"ENTER VALUES FOR A, B, C";A,B,C
30 LET X=-B/(2*A)
40 LET Y=-(B*B-4*A*C)/(4*A)
50 IF A<0 THEN 80
60 PRINT"FUNCTION HAS A MIN. VALUE OF "Y" WHEN X = "X
70 GOTO 90
80 PRINT"FUNCTION HAS A MAX. VALUE OF "Y" WHEN X = "X
90 END
```

Problem 4

```
10 PRINT"DETERMINE ROOTS OF A(X^2)+BX+C = 0"
20 INPUT"ENTER VALUES FOR A, B, C";A,B,C
30 LET D=(B*B)-(4*A*C)
40 IF D<0 THEN PRINT"EQUATION HAS NO REAL ROOTS":GOTO 10
50 LET X=-B/(2*A)
60 IF D=0 THEN PRINT"EQUATION HAS ONE REAL ROOT:X = "X:
                                                 GOTO 10
70 LET X1=(-B+(D^.5))/(2*A):X2=(-B-(D^.5))/(2*A)
80 PRINT"EQUATION HAS TWO REAL ROOTS: X1="X1", X2="X2
90 GOTO 10
100 END
```

APPENDIX III

STUDENT OPINIONNAIRE (A)

STUDENT OPINIONNAIRE (B)

Student Opinionnaire (A)

Express your feeling with regard to Math 20 course just completed.

School.................................Male.....Female.....

1. I was able to do most of the exercises by myself...SA A U D SD
2. I had trouble with most of the lessons.............SA A U D SD
3. We did too many exercises which seemed boring......SA A U D SD
4. It bothers me to make a mistake in mathematics.....SA A U D SD
5. The applications that the teacher introduced were especially interesting...............................SA A U D SD
6. The teacher assigned too much work in the class....SA A U D SD
7. I would have liked more help from the teacher......SA A U D SD
8. I was able to do mathematics without much coaxing by the teacher..................................SA A U D SD
9. I was able to learn much mathematics during this term...SA A U D SD
10. The teacher is good at explaining mathematics......SA A U D SD
11. The teacher assigned too much homework.............SA A U D SD
12. Many of the exercises took too much time to understand thoroughly..............................SA A U D SD
13. I see little use of this type of mathematics.......SA A U D SD
14. The mathematics period seemed too long.............SA A U D SD
15. There are too many repetitive exercises in mathematics.......................................SA A U D SD

16. After studying mathematics I am interested in acquiring further mathematical knowledge...........SA A U D SD

17. I didn't get enough individual attention...........SA A U D SD

18. The teacher checked my homework sufficiently.......SA A U D SD

19. The way we did mathematics gave me an umcomfortable experiences..........................SA A U D SD

20. I now feel better about mathematics than I did before..........................SA A U D SD

21. In Math 20, I gained confidence of doing mathematics......................................SA A U D SD

22. I like doing mathematics..........................SA A U D SD

23. I plan to take mathematics next year...............SA A U D SD

24. Computer programming is very mathematical..........SA A U D SD

25. Computer programming seems too different from mathematics...................................SA A U D SD

26. I didn't like to make a mistake when I did the programming exercises in mathematics...........SA A U D SD

27. I would have liked to have had more explanation in the classroom about programming before I had gone to the microcomputer lab.....................SA A U D SD

28. I would rather have done mathematics in the classroom in the usual way, without programming exercises...................................SA A U D SD

29. I liked working from the BASIC Manual..............SA A U D SD

30. Programming exercises in mathematics gave me a more meaningful way of doing mathematics..........SA A U D SD

31. I would have learned just as much mathematics in the classroom without programming exercise periods.SA A U D SD

32. The BASIC Manual was too hard to understand........SA A U D SD

33. I liked working from problem sheets...............SA A U D SD

34. I was frustrated when I did programming exercises in mathematics.........................SA A U D SD

35. The short review about BASIC by the teacher before each lesson was useful......................SA A U D SD

36. Later programming exercises in mathematics periods were not as much fun as the earlier ones...SA A U D SD

37. The programming exercises in mathematics periods made me more sure of myself when I returned to the classroom....................................SA A U D SD

38. Programming exercises in mathematics helped me to use mathematical knowledge to solve problems....SA A U D SD

39. Other comments (Please indicate, if any.)

....

....

....

....

....

....

Student Opinionnaire (B)

Express your feeling with regard to Math 20 course just completed.

School..................................Male.....Female......

1. I was able to do most of the exercises by myself...SA A U D SD
2. I had trouble with most of the lessons.............SA A U D SD
3. We did too many exercises which seemed boring......SA A U D SD
4. It bothers me to make a mistake in mathematics.....SA A U D SD
5. The applications that the teacher introduced were especially interesting.............................SA A U D SD
6. The teacher assigned too much work in the class....SA A U D SD
7. I would have liked more help from the teacher......SA A U D SD
8. I was able to do mathematics without much coaxing by the teacher...............................SA A U D SD
9. I was able to learn much mathematics during this term.......................................SA A U D SD
10. The teacher is good at explaining mathematics......SA A U D SD
11. The teacher assigned too much homework.............SA A U D SD
12. Many of the exercises took too much time to understand thoroughly..............................SA A U D SD
13. I see little use of this type of mathematics.......SA A U D SD
14. The mathematics period seemed too long.............SA A U D SD
15. There are too many repetitive exercises in mathematics.......................................SA A U D SD

16. After studying mathematics I am interested in acquiring further mathematical knowledge...........SA A U D SD

17. I didn't get enough individual attention...........SA A U D SD

18. The teacher checked my homework sufficiently.......SA A U D SD

19. The way we did mathematics gave me an umcomfortable eriences..........................SA A U D SD

20. I now feel better about mathematics than I did before.................................SA A U D SD

21. In Math 20, I gained confidence of doing mathematics.......................................SA A U D SD

22. I like doing mathematics..........................SA A U D SD

23. I plan to take mathematics next year...............SA A U D SD

24. I would like to have some experience in computer programming.......................................SA A U D SD

25. I would like to know more about computer programming and the application of mathematics.....SA A U D SD

26. A mathematics course should involve computer programming.......................................SA A U D SD

27. Other comments (Please indicate, if any.)

....

....

....

....

....

....

APPENDIX IV

MATH 2Ø TEST

MATH 20

JUNE, 1984 FINAL

Time: 2 h

Name: ________________

Total Marks: 103

Teacher: ________________

Number of pages: 8

INSTRUCTIONS TO STUDENTS

This examination consists of two parts:

Section A: LONG ANSWER QUESTIONS ...43 marks
Partial marks will be given in this section. SHOW YOUR WORK!

Section B: MULTIPLE CHOICE ...60 marks

No partial marks will be awarded. Each question is worth 2 marks. Use the ANSWER SHEET (the last page of this booklet). Indicate your choice of answer by placing an "X" through the appropriate letter. Remove the answer sheet for convenience and place it in the booklet when you are finished.

CALCULATORS ARE PERMITTED

TRIG TABLES ARE ATTACHED

MATHEMATICS 20 FINAL - JUNE, 1984

1. In simplified form $\frac{3\sqrt{2} - \sqrt{3}}{4\sqrt{3} - \sqrt{2}}$ equals

   A. $\frac{11\sqrt{6} - 6}{10}$

   B. $\frac{11\sqrt{6} - 18}{10}$

   C. $\frac{11\sqrt{6} - 6}{46}$

   D. $\frac{11\sqrt{6} - 18}{46}$

2. If $P(x) = x^3 - 6x^2 - 7x$, then the zeros of the polynomial are

   A. 0, -1, 7
   B. 0, 1, -7
   C. -1, 7
   D. 1, -7

3. If $F(x) = -2x^3 + 4x^2 - 17$, then $F(-2)$ equals

   A. -49
   B. -17
   C. 1
   D. 15

4. One factor of $(x-3)^3 - 1$ is

   A. $x^2 + 4x + 16$
   B. $x^2 - 8x + 15$
   C. $x^2 - 7x + 13$
   D. $x^2 - 5x + 7$

5. The slope of a line parallel to $x = 7$ is

   A. 1
   B. -1
   C. undefined
   D. 0

6. The slope of a line perpendicular to $2x - 3y + 7 = 0$ is

   A. $\frac{3}{2}$

   B. $-\frac{3}{2}$

   C. $\frac{2}{3}$

   D. $-\frac{2}{3}$

7. The equation of a line passing through $(2, -\frac{5}{2})$ and parallel to the x - axis is

   A. $2y + 5 = 0$
   B. $2y - 5 = 0$
   C. $x - 2 = 0$
   D. $x + 2 = 0$

8. The equation of the line passing through (4, -2) and having an x-intercept of 3 is

A. $2x + y - 6 = 0$
B. $x + 2y - 3 = 0$
C. $2x + y - 3 = 0$
D. $x - 2y + 6 = 0$

9. The mean for the following set of values $\{34, 53, 13, 41, 34, 47\}$ is

A. 45
B. 34
C. 37
D. 37.5

10. The relation which is NOT a function is

A. $\{(-1, 2), (0,0), (1,-2)\}$
B. $\{(-3,3), (0,0), (-3,-3)\}$
C. $y = -x^2$
D. $y = 4$

11. If $f(x) = x^2 - 1$, then $f(a) - f(a+1)$ is equal to

A. $-2a - 1$
B. 1
C. -1
D. $2(a^2 + 1)$

12. If $G = \{(-2,-4), (0,0), (3,6)\}$, the domain of the inverse of G is

A. $\{-\frac{1}{4}, 0, \frac{1}{6}\}$
B. $\{-\frac{1}{2}, 0, \frac{1}{3}\}$
C. $\{-2, 0, 3\}$
D. $\{-4, 0, 6\}$

13. For $y = -\sqrt{x - 16}$, the range is

A. $\{y \in R,\ y \geq 0\}$
B. $\{x \in R,\ x = 4\}$
C. $\{y \in R,\ y \leq 0\}$
D. $\{x \in R,\ x \geq 16\}$

14. If the roots of a quadratic equation are -2 and $\frac{3}{5}$, then a possible equation is

a. $3x^2 - x - 10 = 0$
B. $5x^2 + 7x - 6 = 0$
C. $2x^2 - 3x = 5$
D. $5x^2 - 7x = 6$

15. The minimum value of $f(x) = 3x^2 + 6x - 4$ is

A. -7
B. -1
C. 1
D. $-\frac{7}{3}$

16. The function $y = 5x^2 - 4x - 1$ is NEGATIVE for

A. $x < -\frac{9}{5}$ or $x > \frac{2}{5}$

B. $-\frac{9}{5} < x < \frac{2}{5}$

C. $-\frac{1}{5} < x < 1$

D. $x < -\frac{1}{5}$ or $x > 1$

17. The equation $\frac{2x-1}{x+3} = \frac{x+3}{2x+1}$ has

A. no real roots
B. one real root
C. two real and unequal roots
D. three real and unequal roots

18. A certain quadratic function has a maximum value of 5 and has its axis of symmetry given by the equation $x = 2$. One possible equation that satisfies these conditions is

A. $f(x) = -4(x-5)^2 + 2$

B. $f(x) = 3(x+2)^2 - 5$

C. $f(x) = \frac{1}{3}(x + 5)^2 - 2$

D. $f(x) = -\frac{3}{4}(x-2)^2 + 5$

19. The solution set for the system $x + 4y = -2$ and $y = 0$ is

A. $(-2, 0)$
B. $y = 0$
C. $\emptyset$ (or the empty set)
D. $x + 4y = -2$

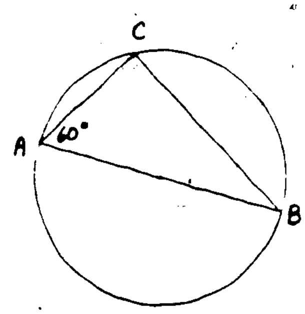
20. The solution set of the system $\begin{cases} x-3y=0 \\ 5y=0 \end{cases}$ is represented by

A. one point
B. two points
C. a line
D. $\emptyset$ (or the empty set)

21. James and Mary bought a lottery ticket and won \$500 000. Since they paid different amounts for the winning ticket, James received \$50 000 more than twice what Mary received. James received

A. \$100 000
B. \$150 000
C. \$300 000
D. \$350 000

22. In the diagram at the right $\overline{AB}$ is a diameter with measure 2. If $\angle CAB = 60^\circ$ then the measure of $\overline{AC}$ equals



A. 1

B. $\sqrt{2}$

C. $\sqrt{3}$

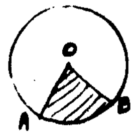
D. 2

23. In a circle of radius 17 cm, a chord is 8 cm from the centre of the circle. The length of the chord is

A. 34 cm
B. 30 cm
C. 16 cm
D. 15 cm

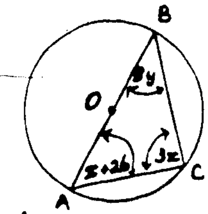
24. For the circle with centre O, /AOB = 72^o, and radius 10 cm the area of the shaded sector, in square cm, is

A. 4π
B. 10π
C. 20π
D. 40π



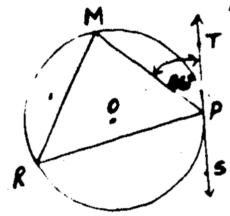
25. The degree measures of /A, /B and /C, respectively are

A. 48, 66, 66
B. 56, 34, 30
C. 56, 94, 30
D. 56, 34, 90



26. For the circle with centre O and tangent TS, if /TPM = 46^o then /MRP equals

A. 23^o
B. 44^o
C. 46^o
D. 92^o

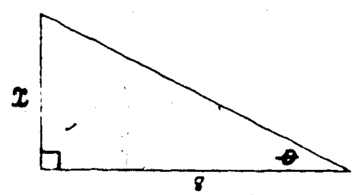


27. Given sin θ = $\frac{4}{5}$ then a value for sec θ could be

A. $\frac{3}{5}$
B. $\frac{4}{5}$
C. $\frac{5}{4}$
D. $\frac{5}{3}$

28. In the diagram to the right, the value of x is

A. 8 sin θ
B. $\frac{8}{\cos\theta}$
C. 8 tan θ
D. 8 cos θ



29. The volume (V) of a cylinder varies jointly with respect to its length (L) and its cross-sectional area (A). If k is a constant of proportionality, the correct equation describing this relationship is

A. $V = k(L+A)$
B. $V = kLA$
C. $V = \frac{kL}{A}$
D. $V = \frac{kA}{L}$

30. The gravitational force of attraction (F) between two objects varies inversely with respect to the square of the distance (d) between them. If the distance between two masses is tripled, the new force between them will be

A. 3F
B. $\frac{F}{3}$
C. 9F
D. $\frac{F}{9}$

LONG ANSWERS

1. Determine the solution set for $\sqrt{4 - 3x} - x = 12$.

(4)

2. Simplify and state the non-permissible values:

$$\frac{5}{2a^2-7a-4} + \frac{3}{2a^2+7a+3} - \frac{2}{a^2-a-12}$$

(4)

3. A triangle has vertices A(-3,-1), B(4,3) and C(11,-6). Determine the length of the median from B to $\overline{AC}$.

(4)

4. If $f(x) = \frac{2x+3}{2}$, then evaluate $f^{-1}(-1)$.

(3)

5. Determine the value of two numbers with a difference of 6 if the sum of their squares is to be a minimum.

(4)

6. Gonzo is planning a motorcycle trip into the mountains. He plans to take 3 hours longer going up the mountains (a distance of 400 km) than coming down (a distance of 300 km). To accomplish this, he must travel 10km/h faster coming down. How long will the entire trip take?

(5)

7. Solve the following systems:

a) $3x - 2y = 6$

$y = \frac{3}{2}x + 3$

(3)

b) $x^2 = -y^2 + 10$

$y = x^2 + 2$

(4)

8. For the circle with centre O, determine the values for the variables indicated.

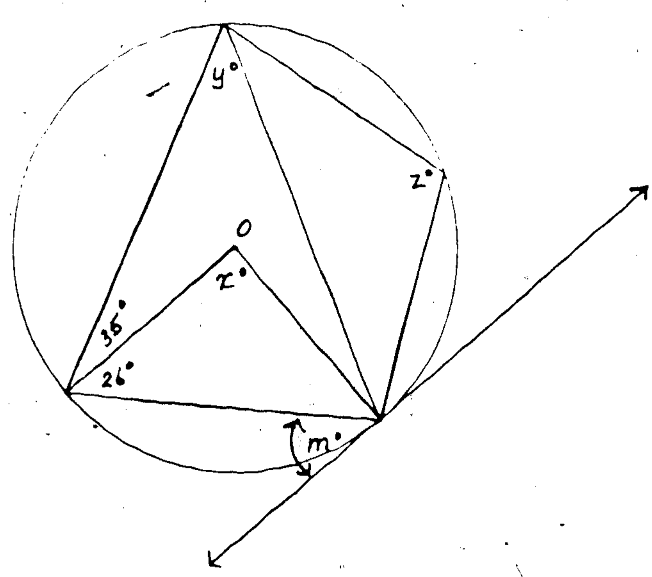
x = ______

y = ______

z = ______

m = ______

(4)



9. From a helicopter the pilot sights two forest fires, both directly west, at angles of depression of 42^{o} and 5^{o}. If the helicopter is flying at an altitude of 200 m how far apart, to the nearest km, are the forest fires?

(4)

10. a) Complete the following frequency distribution table by determining the class boundaries and the relative frequencies.

(2)

Class Boundaries	Class Mark	Frequency	Class Mark X Frequency	Relative Frequency
	54	3	162	
	61	7	427	
	68	8	544	
	75	2	150	
TOTAL	-	20	-	1.00

b) Determine the median for the above data.

(2)

TRIGONOMETRIC RATIOS: SIX FUNCTIONS

DEGREES	SIN	COS	TAN	CSC	SEC	COT
0	.0000	1.0000	.0000		1.0000	
1	.0175	.9998	.0175	57.30	1.0002	57.29
2	.0349	.9994	.0349	28.65	1.0006	28.64
3	.0523	.9986	.0524	19.11	1.0014	19.08
4	.0698	.9976	.0699	14.34	1.0024	14.30
5	.0872	.9962	.0875	11.474	1.0038	11.430
6	.1045	.9945	.1051	9.5668	1.0055	9.5144
7	.1219	.9925	.1228	8.2055	1.0075	8.1443
8	.1392	.9903	.1405	7.1853	1.0098	7.1154
9	.1564	.9877	.1584	6.3925	1.0125	6.3138
10	.1736	.9848	.1763	5.7588	1.0154	5.6713
11	.1908	.9816	.1944	5.2408	1.0187	5.1446
12	.2079	.9781	.2126	4.8097	1.0223	4.7046
13	.2250	.9744	.2309	4.4454	1.0263	4.3315
14	.2419	.9703	.2493	4.1336	1.0306	4.0108
15	.2588	.9659	.2679	3.8637	1.0353	3.7321
16	.2756	.9613	.2867	3.6280	1.0403	3.4874
17	.2924	.9563	.3057	3.4203	1.0457	3.2709
18	.3090	.9511	.3249	3.2361	1.0515	3.0777
19	.3256	.9455	.3443	3.0716	1.0576	2.9042
20	.3420	.9397	.3640	2.9238	1.0642	2.7475
21	.3584	.9336	.3839	2.7904	1.0711	2.6051
22	.3746	.9272	.4040	2.6695	1.0785	2.4751
23	.3907	.9205	.4245	2.5593	1.0864	2.3559
24	.4067	.9135	.4452	2.4586	1.0946	2.2460
25	.4226	.9063	.4663	2.3662	1.1034	2.1445
26	.4384	.8988	.4877	2.2812	1.1126	2.0503
27	.4540	.8910	.5095	2.2027	1.1223	1.9626
28	.4695	.8825	.5317	2.1301	1.1326	1.8807
29	.4848	.8746	.5543	2.0627	1.1434	1.8040
30	.5000	.8660	.5774	2.0000	1.1547	1.7321
31	.5150	.8572	.6009	1.9416	1.1666	1.6643
32	.5299	.8480	.6249	1.8871	1.1792	1.6003
33	.5446	.8387	.6494	1.8361	1.1924	1.5399
34	.5592	.8290	.6745	1.7883	1.2062	1.4826
35	.5736	.8192	.7002	1.7435	1.2208	1.4281
36	.5878	.8090	.7265	1.7013	1.2361	1.3764
37	.6018	.7986	.7536	1.6616	1.2521	1.3270
38	.6157	.7880	.7813	1.6243	1.2690	1.2799
39	.6293	.7771	.8098	1.5890	1.2868	1.2349
40	.6428	.7660	.8391	1.5557	1.3054	1.1918
41	.6561	.7547	.8693	1.5243	1.3250	1.1504
42	.6691	.7431	.9004	1.4945	1.3456	1.1106
43	.6820	.7314	.9325	1.4663	1.3673	1.0724
44	.6947	.7193	.9657	1.4396	1.3902	1.0355
45	[illegible]	[illegible]	1.00[illegible]	1.4142	1.4142	1.0000
46	.7193	.6947	1.0355	1.3902	1.4396	.9657
47	.7314	.6[illegible]	1.0724	1.3673	1.4663	.9325
48	.7431	.669[illegible]	1.1106	1.3456	1.4945	.9004
49	.7547	.656[illegible]	1.1504	1.3250	1.5243	.8693
50	.766[illegible]	.6428	1.19[illegible]	1.3054	1.5557	.8391
51	.7771	.6293	1.2349	1.2868	1.5890	.8098
52	.7880	.6157	1.2799	1.2690	1.6243	.7813
53	.7986	.60[illegible]8	1.3270	1.2521	1.6616	.7536
54	.80[illegible]	.5878	1.3764	1.2361	1.7013	.7265
55	.8192	.5736	[illegible].428	1.2208	1.7435	.7002
56	.8290	.5592	1.4826	1.2062	1.7883	.6745
57	.8387	.5446	1.5[illegible]	1.1924	1.8361	.6494
58	.8480	.5299	1.60[illegible]	1.1792	1.8871	.6249
59	.85[illegible]	.5150	1.6643	1.1666	1.9416	.6009
60	.8[illegible]	.50[illegible]	1.732[illegible]	1.1547	2.0000	.5774
61	.8746	.4848	1.8[illegible]	1.1434	2.0627	.5543
62	.88[illegible]	.4695	1.88[illegible]	1.1326	2.1300	.5317
63	.8910	.4540	1.96[illegible]	1.1223	2.2027	.5095
64	.8988	.4384	2.[illegible]	1.1126	2.2812	.4877
65	.9063	.4226	2.[illegible]44[illegible]	1.1034	2.3662	.4663
66	.9[illegible]35	.4067	2.[illegible]46[illegible]	1.0946	2.4586	.4452
67	.920[illegible]	.3907	2.355[illegible]	1.0864	2.5593	.4245
68	.9272	.3746	2.47[illegible]	1.0785	2.6695	.4040
69	.9336	.3584	2.605[illegible]	1.0712	2.7904	.3839
70	.9397	.34[illegible]	2.747[illegible]	1.0642	2.9238	.3640
71	.9455	.32[illegible]	2.9042	1.0576	3.0716	.3443
72	.9511	.3090	3.0777	1.0515	3.2361	.3249
73	.9563	.2924	3.2709	1.0457	3.4203	.3057
74	.9613	.2756	3.4874	1.0403	3.6280	.2867
75	.9659	.2588	3.732	1.0353	3.8637	.2679
76	.9703	.2419	4.0108	1.0306	4.1336	.2493
77	.9744	.2250	4.3315	1.0263	4.4454	.2309
78	.9781	.2079	4.7046	1.0223	4.8097	.2126
79	.9816	.1908	5.1446	1.0187	5.2408	.1944
80	.9848	.1736	5.6713	1.0154	5.7588	.1763
81	.9877	.1564	6.3138	1.0125	6.3925	.1584
82	.9903	.1392	7.1154	1.0098	7.1853	.1405
83	.9925	.1219	8.1443	1.0075	8.2055	.1228
84	.9945	.1045	9.5144	1.0055	9.5668	.1051
85	.9962	.0872	11.43	1.0038	11.474	.0875
86	.9976	.0698	14.30	1.0024	14.34	.0699
87	.9986	.0523	19.08	1.0014	19.11	.0524
88	.9994	.0349	28.64	1.0006	28.65	.0349
89	.9998	.0175	57.29	1.0002	57.30	.0175
90	1.0000	.0000		1.0000		.0000

MATH MULTIPLE CHOICE ANSWER SHEET

NAME:________________ BLOCK:______ TEACHER:________________

RAW SCORE:____ PERCENT:____

INSTRUCTIONS: Indicate your answer with an X, e.g., A B X D

1. A B C D
2. A B C D
3. A B C D
4. A B C D
5. A B C D

6. A B C D
7. A B C D
8. A B C D
9. A B C D
10. A B C D

11. A B C D
12. A B C D
13. A B C D
14. A B C D
15. A B C D

16. A B C D
17. A B C D
18. A B C D
19. A B C D
20. A B C D

21. A B C D
22. A B C D
23. A B C D
24. A B C D
25. A B

26. A B C D
27. A B C D
28. A B C D
29. A B C D
30. A B C D

APPENDIX V

PROBLEM SOLVING TEST

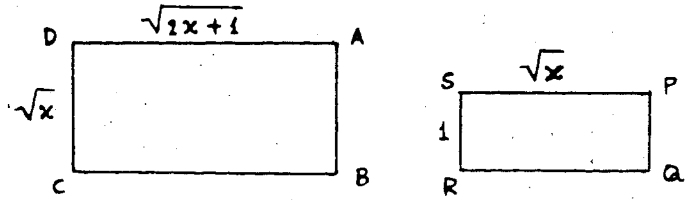
WITH

SCORING PROCEDURE

Problem Solving Test

Problem #1: (Radicals)

Two rectangles have dimensions in metres expressed as radical expressions.



If the difference in the lengths of their diagonals is 2, find the dimensions of the rectangles.

Problem #2: (Linear Equations)

A track for running footraces is in the shape of a rectangle with a semicircle on each end. The length of the rectangle is 10 times the width. What is the length and width of the rectangle if the distance around the track is to be 1500 metres? (π = 3.14)

Problem #3: (System of Equations)

The unit's digit of a two-digit number is 2 greater than 3 times the ten's digit. If 4 is added to the original number, this sum is equal to 4 times the unit's digit. Find the original number.

Problem #4: (Quadratic Functions)

A restaurant makes a profit of $100 per day per table when there are 30 tables in use. When more tables are added, the average profit per table per day decrease by $2 for each extra table. What is the total number of tables that will provide the maximum daily profit? What is the maximum profit?

Problem #5: (Trigonometry)

A pilot and his copilot are flying at an altitude of 1732 feet directly above a tower. The pilot measures the angles of depression to a tower due east to be 30 degrees while the copilot measures the angle of depression to a tower due west to be 60 degrees. How far apart are the two towers that they sighted? ($\sqrt{3}$ = 1.732)

General Scoring Procedure for Problem Solving Test

1. Students are awarded the maximum scores if the correct answer is given, and only correct answers receive maximum scores.

2. Students are awarded partial scores for work which could lead to a correct solution but was incomplete or contained an error.

3. Students are awarded partial scores for showing an understanding of the problem.

4. Students are awarded partial scores for making correct statements about the problem but not discovering a method of obtaining the correct solution.

5. Students are awarded no scores for incorrect statements about the problem or not responding to the problem.

Scoring Procedure for Problem Solving Test

Problem #1 (6 marks)

The length of diagonal AC, in metres, is given by

$$AC^2 = (\sqrt{2x + 1})^2 + (\sqrt{x})^2$$
$$= 2x + 1 + x$$
$$= 3x + 1$$
$$AC = \sqrt{3x + 1}$$

[1 mark for this equation]

The length of diagonal QS, in metres, is given by

$$QS = (\sqrt{x})^2 + (1)^2$$
$$= x + 1$$
$$QS = \sqrt{x + 1}$$

[1 mark for this equation]

Thus, we obtain an equation

$$\sqrt{3x + 1} - \sqrt{x + 1} = 2$$

[2 marks; if this equation is obtained]

$$\sqrt{3x + 1} = 2 + \sqrt{x + 1}$$
$$3x + 1 = 4 + 4\sqrt{x + 1} + (x + 1)$$
$$2x - 4 = 4\sqrt{x + 1}$$
$$x - 2 = 2\sqrt{x + 1}$$

[1/2 mark if student has the above simplified equation]

$$x^2 - 4x + 4 = 4(x + 1)$$
$$x^2 - 4x + 4 = 4x + 4$$
$$x^2 - 8x = 0$$

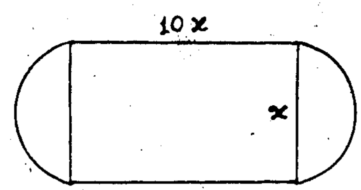
[1/2 mark if student has the above simplified equation]

$$x = 0 \text{ or } x - 8 = 0$$
$$x = 8$$

[1 mark; if this solution is obtained]

Thus, using x = 8, the dimensions of ABCD are $2\sqrt{2}$ m.X$\sqrt{17}$ m. and of PGRS are 1 m.X $2\sqrt{2}$ m.

Problem #2 (6 marks)



[1 mark for the diagram]

Let x be the width

The length is 10x

[2 marks; 1 mark for each unknown]

$10x + 10x + 2\pi(x/2) = 1500$

[2 marks; if this equation is obtained]

$20x + \pi x = 1500$

$(20 + \pi)x = 1500$

$(23.14)x = 1500$

$x = 1500/23.14$

$x = 64.82$

[1 mark; if this solution is obtained]

Thus, the width and the length of the rectangle are 64.82 m. and 648.2 m., respectively.

Problem #3 (6 marks)

Let x be the 10's digit

Let y be the 1's digit

[1 mark for naming the two unknowns]

$$y = 3x + 2$$

$$10x + y + 4 = 4y$$

[2 marks; 1 mark for each equation]

$$10x + 3x + 2 + 4 = 4(3x + 2)$$

$$13x + 6 = 12x + 8$$

$$x = 2$$

[1 mark]

$$y = 3(2) + 2$$

$$y = 8$$

[1 mark]

Therefore, the two-digit number is 28.

[1 mark; if the solution is obtained]

Problem #4 (6 marks)

Let x be number of extra tables that will provide the maximum daily profit

[1 mark for naming the unknown]

Thus, the average profit per table per day decreases by $2x, and the total number of tables is 30 + x

[1 mark; if the above information is obtained]

Maximum daily profit $= (30 + x)(100 - 2x)$ [2 marks]

$= 3000 + 40x - 2x^2$

$= -2(x^2 - 20x - 1500)$

$= -2(x^2 - 20x + 100 - 100 - 1500)$

$= -2[(x - 10)^2 - 1600]$

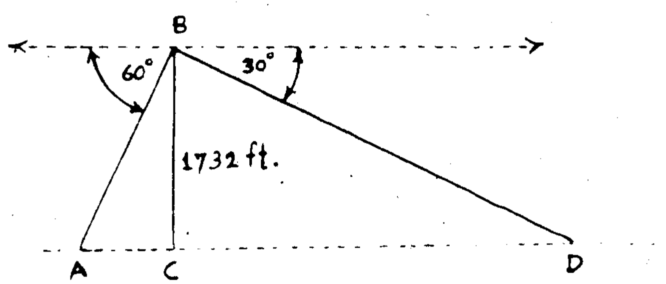
$= -2(x - 10)^2 + 3200$ [1 mark]

When $x = 10$, provides the maximum daily profit.

Hence, the maximum daily profit, \$3200, is obtained when the number of tables is 40.

[1 mark; if the solution is obtained]

Problem #5 (6 marks)



[1 mark for the diagram]

$B\hat{A}C = 60°$

$\tan 60° = BC/AC$ [1 mark]

$\tan 60° = \sqrt{3}$ and $BC = 1732$ ft.

$AC = BC/\tan 60°$

$AC = 1732/\sqrt{3}$

$= 1732/1.732$

$AC = 1000$ ft. [1 mark]

$B\hat{D}C = 30°$

$\tan 30° = BC/CD$ [1 mark]

$CD = BC/\tan 30^\circ$

$\tan 30^\circ = 1/\sqrt{3}$ and $BC = 1732$ ft.

$CD = 1732/(1/\sqrt{3})$

$CD = 1732 \times 1.732$

$CD = 2999.82$ ft. [1 mark]

$AD = AC + CD$

$AD = 1000 + 2999.82$

$= 3999.82$ ft.

Thus, the two towers are 3999.82 feet apart.

[1 mark; if the solution is obtained]

APPENDIX VI

SCHOOL SUBJECTS ATTITUDE SCALES

Subject Rated

Please place only one mark between each pair of words. Be sure not to leave out any of the pairs.

	very much	a bit	neither	a bit	very much	
nice	===	===	===	===	===	awful
boring	===	===	===	===	===	interesting
unpleasant	===	===	===	===	===	pleasant
dislike	===	===	===	===	===	like
bright	===	===	===	===	===	dull
dead	===	===	===	===	===	alive
lively	===	===	===	===	===	listless (inactive, lazy)
exciting	===	===	===	===	===	tiresome (makes a person feel tired)
useless	===	===	===	===	===	useful
important	===	===	===	===	===	unimportant
impractical	===	===	===	===	===	practical (useful or workable)
worthless	===	===	===	===	===	valuable
helpful	===	===	===	===	===	unhelpful
unnecessary	===	===	===	===	===	necessary
harmful	===	===	===	===	===	advantageous (brings good or gain)
meaningful	===	===	===	===	===	meaningless (doesn't make sense)
hard	===	===	===	===	===	easy
light	===	===	===	===	===	heavy (a lot of work)
clear	===	===	===	===	===	confusing (mixes a person up)
complicated	===	===	===	===	===	simple
elementary	===	===	===	===	===	advanced (beyond the beginning level)
strange	===	===	===	===	===	familiar
understandable	===	===	===	===	===	puzzling (hard to understand)
undemanding	===	===	===	===	===	rigorous (has to be exactly right)

VITA

NAME: Somchai Chuchat

PLACE OF BIRTH: SinghBuri, Thailand

YEAR OF BIRTH: 1953

POST-SECONDARY EDUCATION AND DEGREES:

Tepsatri Teachers' College
LopBuri, Thailand
1970-1974 Certificate of Education

Srinakharinwiroth University
Bangkok, Thailand
1974-1976 B.Ed.

Chulalongkorn University
Bangkok, Thailand
1976-1978 M.Ed.

HONOURS AND AWARDS:

Shell Thailand Company Scholarship
Tepsatri Teachers' College
1970-1971

Caltex Company Scholarship
Tepsatri Teachers' College
1972-1973

Bangkok Bank Limited Scholarship
Srinakharinwiroth University
1974-1975

RELATED WORK EXPERIENCE:

Lecturer
Srinakharinwiroth University
Bangkok, Thailand
1978-1981

Research Assistant
University of Alberta
1981-1984

Teaching Assistant
University of Alberta
1984-1985

PUBLICATIONS:

"The Usage of Herbartian Method and Learning Theory in Teaching Mathematics", Journal of Mathematics (Published in Thai), Vol. 24, Number 254-255, November-December 1979, pp. 6-9.