INFORMATION TO USERS

This dissertation was produced from a microfilm copy of the original document. While the most advanced technological means to photograph and reproduce this document have been used, the quality is heavily dependent upon the quality of the original submitted.

The following explanation of techniques is provided to help you understand markings or patterns which may appear on this reproduction.

1. The sign or "target" for pages apparently lacking from the document photographed is "Missing Page(s)". If it was possible to obtain the missing page(s) or section, they are spliced into the film along with adjacent pages. This may have necessitated cutting thru an image and duplicating adjacent pages to insure you complete continuity.

2. When an image on the film is obliterated with a large round black mark, it is an indication that the photographer suspected that the copy may have moved during exposure and thus cause a blurred image. You will find a good image of the page in the adjacent frame.

3. When a map, drawing or chart, etc., was part of the material being photographed the photographer followed a definite method in "sectioning" the material. It is customary to begin photoing at the upper left hand corner of a large sheet and to continue photoing from left to right in equal sections with a small overlap. If necessary, sectioning is continued again — beginning below the first row and continuing on until complete.

4. The majority of users indicate that the textual content is of greatest value, however, a somewhat higher quality reproduction could be made from "photographs" if essential to the understanding of the dissertation. Silver prints of "photographs" may be ordered at additional charge by writing the Order Department, giving the catalog number, title, author and specific pages you wish reproduced.

University Microfilms
300 North Zeeb Road
Ann Arbor, Michigan 48106
A Xerox Education Company
JURICK, Robert Rudolph, 1938-
COMPUTER-ORIENTED INSTRUCTIONAL SYSTEM FOR
TEACHING ANALYTIC GEOMETRY.
The Ohio State University, Ph.D., 1972
Mathematics

University Microfilms, A XEROX Company, Ann Arbor, Michigan
COMPUTER-ORIENTED INSTRUCTIONAL SYSTEM
FOR TEACHING ANALYTIC GEOMETRY

DISSERTATION
Presented in Partial Fulfillment of the Requirements for
the Degree Doctor of Philosophy in the Graduate
School of The Ohio State University

By
Robert Rudolph Jurick, B.S., M. Ed., M.S.

The Ohio State University
1972

Approved by

[Signature]
Adviser
College of Education
PLEASE NOTE:

Some pages may have
indistinct print.
Filmed as received.

University Microfilms, A Xerox Education Company
ACKNOWLEDGEMENTS

My sincere appreciation is extended to my adviser, Dr. Harold C. Trimble, for the encouragement and orientation he provided throughout this study. His thoughtful questions and suggestions have helped me clarify my goals and keep the entire effort in proper perspective. In addition his numerous reviews of the drafts have greatly improved the readability of this manuscript.

A debt of gratitude is also owed to the other members of my reading committee, Dr. Richard J. Shumway and Dr. Charles Saltzer: to Dr. Shumway for the insight provided regarding computers in mathematics education and to Dr. Saltzer for providing much of my background in computer science and mathematics.

The cooperation of Cincinnati Country Day School in providing the facilities for this study is appreciated. My sincere gratitude goes to Ralph Klitz who spent many extra hours in coordinating activities and teaching the course. Appreciation is also extended to Jeff Spain for his assistance in writing computer programs and to David Laird for his many suggestions in developing the course material.

I am especially grateful to Peggy Hayes for her patience and dedication in typing this manuscript and to James Hudson for his consideration as my supervisor and his constructive comments on the original drafts. Finally, I would like to dedicate this manuscript to my family, and especially my wife, for their encouragement during the many years of my education.
VITA

April 28, 1938. . . . Born - Canonsburg, Pennsylvania

1959. . . . . . . . B.S.M.E., Carnegie Institute of Technology, Pittsburgh, Pennsylvania


1961-1962. . . . Mathematics Teacher, Shady Side Academy, Pittsburgh, Pennsylvania


1963-1967 . . . . Officer, United States Air Force

1967-1972 . . . . Mathematician, United States Air Force Wright-Patterson Air Force Base, Ohio

1968. . . . . . . M.S., The Ohio State University, Columbus, Ohio

PUBLICATIONS


FIELD OF STUDY

Major Field: Mathematics Education. Dr. Harold C. Trimble
Minor Field: Computer and Information Sciences. Dr. Charles Saltzer
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>ACKNOWLEDGEMENTS</th>
<th>ii</th>
</tr>
</thead>
<tbody>
<tr>
<td>VITA</td>
<td>iii</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>vi</td>
</tr>
<tr>
<td>LIST OF ILLUSTRATIONS</td>
<td>vii</td>
</tr>
</tbody>
</table>

## CHAPTER 1. INTRODUCTION

1.1 Computers in education ............................................. 1
1.2 Problem to be investigated ........................................ 2
1.3 Organization of the report ........................................ 3
1.4 Classification of computer usage in mathematics education .................................................. 4
1.5 Motivation for the study .......................................... 7
1.6 Design and implementation of the study .................... 11
1.7 Significance of the study ....................................... 13

## CHAPTER 2. RELATED LITERATURE AND RESEARCH

2.1 Introduction ...................................................... 15
2.2 Terminal presented instruction ................................. 15
2.3 Instructional management systems ......................... 29
2.4 Mathematics information systems .......................... 38
2.5 Computational tool ............................................. 43
2.6 Evaluation of algorithms .................................... 45
2.7 Demonstration of mathematical concepts or principles. 53
2.8 Computer as a subject of instruction ..................... 58
2.9 Integrated systems ............................................. 61
2.10 Summary ........................................................... 63

## CHAPTER 3. DESIGN OF STUDY

3.1 Introduction ..................................................... 66
3.2 Environment ..................................................... 66
3.3 Students ........................................................... 69
3.4 Organization of the course .................................... 70
3.5 Computer activities ............................................ 71
3.6 Daily operating procedures ................................... 83
3.7 Tests and evaluation ........................................... 85
CHAPTER 4. IMPLEMENTATION

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1 Introduction.</td>
<td>87</td>
</tr>
<tr>
<td>4.2 Orientation.</td>
<td>87</td>
</tr>
<tr>
<td>4.3 Quiz correction and report generation</td>
<td>90</td>
</tr>
<tr>
<td>4.4 Homework checking.</td>
<td>94</td>
</tr>
<tr>
<td>4.5 TPI programming.</td>
<td>103</td>
</tr>
<tr>
<td>4.6 Student record keeping.</td>
<td>110</td>
</tr>
<tr>
<td>4.7 Daily operating procedures.</td>
<td>116</td>
</tr>
<tr>
<td>4.8 Tests and evaluation.</td>
<td>117</td>
</tr>
</tbody>
</table>

CHAPTER 5. REPORT OF DATA

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1 Introduction.</td>
<td>121</td>
</tr>
<tr>
<td>5.2 Pretest.</td>
<td>124</td>
</tr>
<tr>
<td>5.3 Quizzes.</td>
<td>125</td>
</tr>
<tr>
<td>5.4 Programming.</td>
<td>125</td>
</tr>
<tr>
<td>5.5 Terminal presented instruction.</td>
<td>129</td>
</tr>
<tr>
<td>5.6 Final exam.</td>
<td>129</td>
</tr>
<tr>
<td>5.7 Correlation analysis.</td>
<td>133</td>
</tr>
<tr>
<td>5.8 Achievement.</td>
<td>137</td>
</tr>
</tbody>
</table>

CHAPTER 6. SUMMARIZATION AND FUTURE RESEARCH

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1 Introduction.</td>
<td>138</td>
</tr>
<tr>
<td>6.2 Limitations.</td>
<td>139</td>
</tr>
<tr>
<td>6.3 Observations.</td>
<td>141</td>
</tr>
<tr>
<td>6.4 Recommendations for future classroom implementations</td>
<td>151</td>
</tr>
<tr>
<td>6.5 Suggestions for future comparative evaluations</td>
<td>168</td>
</tr>
<tr>
<td>6.6 Recommendations for future research and development</td>
<td>174</td>
</tr>
</tbody>
</table>

BIBLIOGRAPHY.                             | 181  |

APPENDIXES (Separate table of contents). | 197  |
### LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Course outline: organized by sections of the text</td>
<td>71</td>
</tr>
<tr>
<td>2.</td>
<td>Chronology of significant events</td>
<td>88</td>
</tr>
<tr>
<td>3.</td>
<td>BASIC statements for generating random input data to homework problem no. 9</td>
<td>98</td>
</tr>
<tr>
<td>4.</td>
<td>Overall student summary</td>
<td>122</td>
</tr>
<tr>
<td>5.</td>
<td>Correlation analysis</td>
<td>134</td>
</tr>
<tr>
<td>6.</td>
<td>Comparison of program completion to final exam scores</td>
<td>135</td>
</tr>
<tr>
<td>7.</td>
<td>Comparison of TPI program usage to final exam scores</td>
<td>136</td>
</tr>
<tr>
<td>8.</td>
<td>Comparison of results on similar pretest and final exam questions</td>
<td>137</td>
</tr>
</tbody>
</table>
# LIST OF ILLUSTRATIONS

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Simulated dialogue with a mathematics information system.</td>
<td>39</td>
</tr>
<tr>
<td>2.</td>
<td>Sample quiz card.</td>
<td>72</td>
</tr>
<tr>
<td>3.</td>
<td>Questions from TPI No. 1.</td>
<td>74</td>
</tr>
<tr>
<td>4.</td>
<td>Homework assignments for first two sessions</td>
<td>76</td>
</tr>
<tr>
<td>5.</td>
<td>Arrangement of check program.</td>
<td>77</td>
</tr>
<tr>
<td>6.</td>
<td>Procedure for running homework programs</td>
<td>78</td>
</tr>
<tr>
<td>7.</td>
<td>Sample output from check program No. 2.</td>
<td>80</td>
</tr>
<tr>
<td>8.</td>
<td>Teacher daily summary report.</td>
<td>82</td>
</tr>
<tr>
<td>9.</td>
<td>Logic of prescription process</td>
<td>84</td>
</tr>
<tr>
<td>10.</td>
<td>Allocation of central memory</td>
<td>112</td>
</tr>
<tr>
<td>11.</td>
<td>Programs per student.</td>
<td>126</td>
</tr>
<tr>
<td>12.</td>
<td>Programs per session.</td>
<td>126</td>
</tr>
<tr>
<td>13.</td>
<td>Number of students programming each day</td>
<td>127</td>
</tr>
<tr>
<td>14.</td>
<td>Tries per homework program</td>
<td>127</td>
</tr>
<tr>
<td>15.</td>
<td>TPI programs tried per student</td>
<td>130</td>
</tr>
<tr>
<td>16.</td>
<td>TPI programs tried per session</td>
<td>130</td>
</tr>
<tr>
<td>17.</td>
<td>Number of students using TPI each day</td>
<td>131</td>
</tr>
<tr>
<td>18.</td>
<td>Tries per TPI program</td>
<td>131</td>
</tr>
</tbody>
</table>
CHAPTER ONE
INTRODUCTION

1.1 Computers in Education

"The computer is the most remarkable machine by far, yet made by man [Snow, 1966]." Indeed, the computer has influenced our society during the 25 years since its inception and much of its impact on our life styles is still to come. It is not inconceivable that by the year 2000 all large-scale manufacturing, transportation, and communication will be completely automated, money will be eliminated, and each dwelling will have a "home-information center" including a small computer linked to a local computer utility and TV sets used for displaying information as well as entertainment. We are now witnessing the embryonic growth of the computer revolution and have the opportunity (and responsibility) to influence its use for the betterment of our society.

Many educational institutions have recognized the mind-extending capabilities of the computer and have included it in their instructional programs. Although the universities were the first to apply computers to education, many secondary schools are choosing this resource for both instructional and administrative data processing applications. Molnar (1971) reports that 34 percent of the secondary schools in the United States have access to and/or use computers. The use of computers in secondary school has been a controversial
issue. One critic, Anthony Oettinger, states:

The schools haven't got any money. Universities, non-profit creatures of the government and private industry haven't got any ideas save the present innovation fad which favors highly visible quickly approaches creating the illusion of progress. No one is able or willing to take time and risks. (Oettinger, 1969, p. 220)

A more favorable opinion is expressed by Glenn Bryan:

I think that the public wants the things that computers can provide to education, such things as:

- remote access;
- around-the-clock availability;
- enduring patience;
- student anonymity;
- self-pacing; and
- some forms of individualized instruction.

(Bryan, 1969, p. 19)

1.2 Problem to be Investigated

Assuming that a school has chosen to incorporate computers into the instructional process, the teacher desiring to use the computer for a specific mathematics course is faced with a difficult problem: the responsibility of utilizing this expensive and limited resource to its greatest advantage. This problem is not lessened by the fact that although much research and development effort has been expended and reported on individual computer activities such as problem solving or drill-and-practice, little emphasis has been given to integrating the best advantages of each activity into an organized mathematical course.

The magnitude of the above problem is emphasized by the following finding from a survey of teachers in Missouri and its adjoining states: "The secondary teachers felt that methodology in utilizing computer time in their mathematics courses was their principal need
1.3 Organization of the Report

Hopefully, this report will be of interest to three principal audiences: 1) teachers and administrators interested in an overview of the design, implementation, and results of computer applications in secondary schools; 2) secondary school teachers who are interested in actually implementing part or all of the design described in this report; and 3) mathematics educators who are interested in developing methodologies for computer usage in the schools. For those interested in a condensed version of the report, summary paragraphs are provided at the beginning of the more detailed sections, particularly those sections concerning the implementation of the study.

In order to incorporate as many of the current computer activities as possible into a course design, it was necessary to do an extensive review of the literature. This review has resulted in a rather long Chapter Two—Related Literature and Research. This chapter has been separated into sections by principal classifications of computer activities. To facilitate reading, each of these sections begins with a short summary. The extensive review of the literature also resulted in a rather long bibliography and therefore it was decided to separate it into two parts: references cited in the text, and references used as background material.

Chapter Three describes the design of the study; Chapter Four describes its implementation; and Chapter Five contains a report of the data collected. Chapter Six contains the author's observations, some searchable questions, recommendations for future implementation of the
course and future replications of the study, and concludes with recommendations for future research and development by teachers, doctoral candidates, professional societies, and computer firms. Mathematics educators will probably be most interested in the author's observations, researchable questions, and recommendations for future research.

Since there are over 80 pages in the appendixes, a separate table of contents on page 199 has been prepared. Much of the material in the appendixes are course items such as quizzes, homework, etc.; however, tables of usage of the course components and a listing of student comments are also included.

1.4 Classification of Computer Usage in Mathematics Education

The creation of new computer applications and the improvement and expansion of the original applications has resulted in large array of conflicting and overlapping titles and acronyms to describe these activities. Salisbury (1971) lists 21 different names in the literature which describe what he defines as Computer-Assisted Instruction (CAI). His article recommends a moratorium on new names and offers a glossary of inclusive, non-overlapping titles that describe computer usage in education. Unfortunately Salisbury's classifications appear to be too general for describing specific mathematics education activities.

The classifications proposed by this investigation are:

- Terminal Presented Instruction (TPI)
- Instructional management systems
- Mathematical information systems
- Computational tool
• Evaluation of algorithms
• Demonstration of mathematical concepts or principles
• Computer as a subject of instruction

Hopefully these activities do not overlap and are exhaustive. They will be briefly described in the following paragraphs.

Terminal Presented Instruction (TPI). This application involves using the computer to replace part or all of the traditional classroom lecture, textbook, and tutorial processes by presenting course material to a student and interacting with him while he is situated at a computer terminal. Although this application is often referred to as CAI, the use of this title was avoided because of possible ambiguities in its meaning.

Instructional management systems. This application incorporates much of the theory of management information systems currently used in many large industrial corporations. It includes the collection and reporting of student data from activities such as quizzes, homework, TPI, etc. and usually also includes procedures for prescribing individualized student activities. If prescription is included in the system, then data on the availability and effectiveness of course resources must also be collected and maintained. A term often used to describe this activity is Computer Managed Instruction or CMI.

Mathematical information systems. This application allows a student to quickly access an information bank containing items such as definitions of mathematical terms, proofs of theorems, dependency relationships between theorems, demonstration problems, formulas, etc. It might also be described as an automated mathematical library with
very intricate cross-referencing schemes.

Computational tool. This application could be described as using the computer as a super desk calculator or slide rule. Its use presupposes that there are situations in mathematics education in which the student is not too concerned about the problem solving process but only wants the correct answer to a problem. Examples of these situations might be the use of the computer for calculating the square root of a number, or multiplying matrices, or generating random numbers, or plotting an exotic function.

Evaluation of algorithms. This application differs from computation in that the student is concerned with the problem solving process, particularly for those problems that can be solved by the use of an algorithm. An algorithm is a problem solving procedure with the following characteristics:

- it is composed of an ordered sequence of operations
- there is no ambiguity at any step in the sequence
- the solution is obtained after a finite number of steps and
- it has some generality.\(^1\)

Algorithms are often used to teach students to recognize the assumptions and interrelationships of mathematical techniques for problem solving; however, since algorithms can be just as incorrect as any other problem solution, some method must be used to evaluate them. This method normally involves coding the algorithm into a computer

\(^1\)An alternative definition by Traktenbrot (1963) is "a list of instructions specifying a sequence of operations which will give the answer to any problem of a given type [p. 3]."
program and having the computer perform the finite (but often very
large) number of steps to reach the solution. The use of a computer
also allows the evaluation of the algorithm for a large subset of the
general class of problems it was designed to solve.

Demonstration of mathematical concepts or principles. This applica-
tion includes the use of computer-driven plotters, graphical displays,
and computer-animated movies to assist a classroom lecturer in illus-
trating mathematical concepts or principles that are difficult to vis-
ualize. Also included in this application is the use of investiga-
tive and simulation programs to demonstrate the behavior of complex
systems. An investigative program is a prewritten program which
asks the student for data and then reports the results of using that
data. "Data" in the above context does not necessarily imply numbers;
it could also include equations, theorems, definitions, etc.

Computer as a subject of instruction. This application is concerned
with teaching the student the principal characteristics of computers
and computer applications. Gaining competency in a computer language
is usually a goal of this application, but also included are activi-
ties that focus attention on the computer's advantages and limita-
tions to better prepare a student to adjust to an increasingly
computerized society. In some schools this application is taught in
courses called Computer Science; in others it is taught in mathema-
tics or science courses that have been computerized.

1.5 Motivation for the Study

During the author's six years of scientific programming before
attending The Ohio State University full-time during the 1969-70
academic year, it was often observed that programming a solution to
a problem forces a complete involvement with that problem. This in-
volvelement usually led to an understanding of problems that the author
had never obtained in college courses. It was truly "self-motivated
learning" which, according to Carl Rogers (1961), is the only type of
learning that is of any consequence.

Another observation often made was that, although all of the
scientific programmers had either bachelor or master degrees in mathe-
matics, very little complicated mathematics was needed to do their
job. If this observation was valid, might not high schools offer
courses in scientific programming to junior and senior mathematics
students to offer them an alternative to college, or a skill for sum-
mer or part-time employment while in college, or a headstart for their
college computer courses?

While at The Ohio State University during the 1969-70 academic
year, an investigation of the literature relating to computers in edu-
cation led to a few more observations. One observation was that edu-
cators were tired of hearing predictions about the great potential of
the computer and were more interested in the current results which
were far from encouraging, especially when cost was considered. Con-
sidering Terminal Presented Instruction, the author wondered if its
development and cost problems could not be reduced by encouraging the
programming of smaller instructional units to be used for tutorial
or advancement work in conjunction with teacher presented courses.

During the brief closing of The Ohio State University in the
spring of 1970, the author visited a few high schools and universities
in the Dayton area and talked with representatives of computer manufacturing companies. These visits were in preparation for a seminar in computer usage in mathematics education to be conducted by the author in the summer of 1970. The seminar was part of a National Science Foundation Summer Institute in Computer Science which was sponsored by the Computer and Information Science Department of The Ohio State University. The criteria for selecting participants was experience with the computer in the classroom or anticipating the installation of a computer in their school within the next year. From conversations with the teachers who were using computers, it quickly became apparent that the majority of them were teaching beginning programming courses and had, at most, meager plans for offering follow-on courses or for integrating the computer into the mathematics or science curriculum. The conversations with the computer manufacturer representatives, however, indicated a growing awareness of the marketing potential for computers in education and even a willingness on the part of the manufacturers to assist in the instructional process.

During the same summer, the author took a course in which he learned both the BASIC and COURSEWRITER languages and, after helping a friend who was programming some mathematics instruction via COURSEWRITER, confirmed his contention that TPI segments should be reduced in length and added the stipulation that they should be easily programmed. The author also observed that, except for report generation capabilities, BASIC was as powerful as Fortran\(^1\) and, in some cases, professional computer systems of the time were not as powerful.

\(^1\)Fortran is the language most commonly used for scientific computation.
more flexible.

Upon returning to work at Wright-Patterson Air Force Base in September, 1970, the author was appointed director of a continuing education course called Computer Capability and Utilization. Experience with computerized quiz and homework grading and reporting during this course was convincing evidence of the potential of the computer for instructional management. The individual student reports seemed to result in more student responsibility and less emphasis upon trying to pass the minimal requirements. The computerized grading process helped the teacher more than the student, but the reporting was useful to both, especially when meaningful summary reports were also generated.

Throughout this growing awareness of the problems facing computer integration into the education process, the author had maintained the conviction that analytic geometry would be an excellent course to try to integrate computer usage. This conviction was based upon the observation that a large subset of the scientific problems solved by computers were either pure analytic geometry problems or required the solution of analytic geometry problems as part of a more complex problem. It was also observed that the pictorial representation of geometric figures, either by computer-generated graphs or displays, seemed to offer penetrating insight into the behavior of the physical phenomenon being studied. More specifically, these pictorial representations seemed to reduce the amount of abstraction necessary to predict the behavior of the geometric figures under translations, rotations, and changes in the values of the parameters of the equations.
Consolidating the above observations, the author concluded that a useful contribution to mathematics education would be to investigate the modification of an established course (analytic geometry) by including the TPI, instructional management, and algorithm evaluation capabilities of the computer as part of the course's instructional organization. The object of this effort would not be to compare the computerized approach to a non-computerized approach but rather to develop course materials and to record the results of this development and its implementation. The recorded results would hopefully eliminate some wasted efforts by future developers and identify some researchable topics for future experimenters.

To attempt this developmental effort it was necessary to find a secondary school with computer facilities that was also teaching analytic geometry. A desirable condition was that the students of the analytic geometry course also be competent programmers.

1.6 Design and Implementation of the Study

The study was conducted at Cincinnati Country Day School in Cincinnati, Ohio. Although this school didn't meet the specified criteria of offering a course in analytic geometry, an arrangement was made with the head of the mathematics department, Mr. Ralph Klitz, to teach analytic geometry during the last month of a sophomore geometry course he was teaching. The 18 students in his course were the best mathematics students of the sophomore class and had had some exposure to BASIC. It was agreed that these students would not be graded on their participation in the study.
The computer available was a minicomputer with limited memory capacity. Input was via a teletypewriter, optical mark card reader, and high speed paper tape reader. Output was either paper tape or printed text from the teletypewriter.

A textbook, "Contemporary Analytic Geometry" (Wade & Taylor, 1969), was chosen for the course and the individual sections of its chapters were used as organizing units. The course consisted of 20 class sessions with a ten-minute quiz beginning each session except the first and the last; the last session was used for a final exam. A pretest and an orientation lecture were given before the course began and a class session was devoted to written student evaluations after the course had ended.

Before the study began, a course schedule was developed by the author and Mr. Klitz. This schedule specified the sections of the text that would be discussed during each lecture. The daily quiz questions were related to sections discussed in the previous day's lecture and the homework assignment to the current day's lecture. The homework assignment for each session consisted of two short programs to be written in BASIC; this requirement resulted in a maximum of 38 programs for the course. In addition, ten TPI programs were made available to the students for remediation or advancement. These were also related to the sections of the text.

The computer was used for grading the quiz questions, checking the correctness of the BASIC programs, and administering the TPI programs. In addition, a record of each student's quiz scores, homework programs completed, and TPI programs tried was permanently stored in
the computer's memory. This information was used to generate a daily teacher's report which included a suggestion for each student's next activity. These suggestions were generated by searching the student records for unlearned sections of the text and then prescribing activities to facilitate the learning of these sections.

1.7 Significance of the Study

It is the author's belief that analytic geometry can be more effectively taught in a secondary school classroom environment if several types of computer activities are properly integrated into the overall course organization. This is in comparison to no computer activities or a single computer activity such as TPI or programming. However, before such a contention can be affirmed, it is necessary to show that many computer activities can indeed be integrated into the overall course organization. Furthermore, assuming that integration is possible, several trial efforts are needed to develop a properly integrated course before a fair comparison can be made.

This study will therefore attempt to 1) establish the existence of an integrated computerized secondary school analytic geometry course and 2) investigate specific strategies to enhance the development of a methodology for proper integration.

Some of the specific strategies that will be investigated are:

1) the use of a minicomputer for course development and implementation

2) the feasibility of teacher development, particularly with respect to manhours required
3) the feasibility of textbook sections as an organizing unit
4) the use of prepunched, pencil-marked cards as a media for quiz data collection
5) the feasibility of short TPI segments, and the feasibility of BASIC as a language for implementing them
6) the feasibility of requiring programs as the only form of homework
7) the feasibility of check programs for verifying program correctness and
8) the use of pencil-marked cards for programming.
CHAPTER TWO
RELATED LITERATURE AND RESEARCH

2.1 Introduction

This chapter will present some current developmental efforts and research findings related to integrating the activities described in Section 1.2 into the mathematics curriculum. These activities are:

- Terminal Presented Instruction (TPI)
- Instructional management systems
- Mathematical information systems
- Computational tool
- Evaluation of algorithms
- Demonstration of mathematical concepts or principles
- Computer as a subject of instruction

After each activity is discussed in a separate section, a section will be devoted to current efforts in combining two or more of these activities. The last section of the chapter will be a brief "state-of-the-art" exposition of computer use in mathematics education; it will include a review of some of the major problems encountered and of some of the questions still unanswered.

2.2 Terminal Presented Instruction (TPI)

TPI is often classified according to the categories: drill-and-practice, tutorial, and dialogue. Drill-and-practice (D&P) presents drill questions to a student depending upon his current learning level and his response to previously asked questions. The typical D&P
session is short (less than five minutes) with little textual information presented and limited student input, i.e., only answers to presented questions. Tutorial TPI usually presents textual (and sometimes pictorial and audio) information to students to supplement or replace a classroom lecture. This mode usually requires longer sessions, more flexible responses, and also presents questions as a function of student response. Dialogue TPI is similar to tutorial TPI except that responses are even more flexible and the student is given much more freedom to alter the instructional sequence. Although typewriter terminals are normally used for D&P, a video display (Cathode Ray Tube or CRT) is usually considered preferable for tutorial and dialogue presentations.

TPI programs are usually organized with respect to preestablished instructional objectives and normally include, or have access to, individual student response data thereby allowing each presentation to be "individualized." The languages used to write TPI programs are called "author" languages and usually require little computer experience to use. Despite the availability of these languages, the time to develop a TPI program is quite extensive, usually in excess of 100 hours of development time for one hour of instruction.

Drill-and-practice programs have been used most predominately in elementary and junior high schools, whereas tutorial TPI has been used mostly in colleges. Dialogue TPI is still in the development stage with very few implementations. Most students and teachers have enjoyed using TPI, but its effectiveness in increasing learning is still questionable.
The major factors inhibiting the introduction of tutorial TPI into secondary schools are its costs, particularly terminal costs, and the availability of qualified course developers. Although costs seem to be an inhibiting factor, it should be noted that the most expensive TPI programs (tutorial and dialogue) are offering instruction in a mode not normally available in secondary schools; i.e., individual tutoring and dialogue with a subject expert.

One alternative to reducing tutorial TPI costs is to use it for parts of courses instead of whole courses. However, there are situations in which the presentation of a complete course via a computer terminal has advantages, namely:

- Adult education
- Job training or retraining
- Home instruction for the handicapped
- Upgrading teacher knowledge by offering in-service courses
- Offering difficult courses in isolated schools which have small teaching staffs of limited preparation
- Offering courses of special interest such as microbiology, anthropology, Serbo-Croatian, Linear Programming, etc. in schools that do not have teachers capable of teaching these courses or in schools whose teachers, although capable, have a full teaching load and therefore could not offer a course just for one or two students.

The remainder of this section will discuss in more detail some of the variables influencing TPI development and implementation and will describe how these variables are handled in a few of the more commonly-known TPI systems. A resume of research on effectiveness and attitudes will also be presented, and the section will conclude with a discussion of the implications of TPI research and development for this study.
Variables influencing TPI development. Some of the major variables influencing TPI development are 1) specification of instructional objectives, 2) instructional strategy, 3) form of presentation, 4) construction of programs, and 5) maintenance of programs. Instruction objectives are also called "behavioral objectives" which are defined by Hansen (1968) to be "hypothesized propositions about what could be and ought to be achieved in the prescribed instructions [p. 73]."

The specification of objectives usually includes determining entry behaviors, interrelationships of objectives, and criteria for determining attainment of these objectives. According to Hansen, "Without a doubt, the process of formulating behavioral objectives is far more complex than portrayed by anyone to date [p. 73]." Although the specification of objectives does help organize a course, regardless of its eventual computerization, one note of warning is presented by Andree (1968):

...I have found that some of our best courses are given by professors whose objectives for a given course are constantly vacillating, not only from year to year, but from week to week. The objectives of a course may well be a function of the current student response and what the teacher happened to be reading last night in a mathematical journal and a lot of other facts [p. 98].

The chosen instructional strategy may be typed as either D&P, tutorial, or dialogue; however considerable flexibility of presentation exists even within these classifications. Some of the options available to course authors include a choice of question format, the type of response allowed (including timed responses), the criteria for determining if a response is correct, the action to be taken on correct or incorrect responses, any required access to a student data bank, and the form of summary printouts to student and teacher.
The selected form of presentation may be either printed or displayed text, graphs, or pictures, and may even include audio accompaniment. The eventual terminal chosen is probably the most limiting factor on the creativity and ingenuity of the course developer and, unfortunately, the choice of terminal is too often dictated by cost. The terminal most commonly used for TPI presentation has been a typewriter\(^1\) connected to the computer; however recently the Cathode Ray Tube (CRT) video-display terminal has become a cost-competitive alternative. The advantage of a CRT to a typewriter is that it is noiseless and it can display a large amount of text almost instantly; its disadvantage is that it lacks permanent copy. McConnell (1968) suggests that typewriters be used for D&P because of the relatively low volume of output and requirements for hard copy, but that CRT's are more preferable for the tutorial and dialogue modes.

Considerable variety is available in terminals for special purposes, although usually at an increase in cost. For students who can't read, the normal touch-tone telephone has been quite effective. For geometric oriented presentations, CRT's with graphic capabilities are available as are hard copy plotters connected to typewriter terminals. For pictorial presentations the plasma display terminal (Alpert & Bitzer, 1970) and computer-controlled television displays are available (Ampex, 1972; Bunderson, 1971). Considerable product development is currently taking place in the terminal industry and, hopefully, will

\(^1\)The most common are the model 33 teletypewriters and IBM 2741 selectric typewriters.
result in future terminals that will no longer constrain the creativity of the course authors.

One of the most critical questions in the integration of terminal presented instruction in the mathematics curriculum is: "Who will write the programs to present the course?" One answer is a team of content scholars, behavioral scientists, full-time curriculum writers, full-time TPI coders, TPI systems programmers, proctors during instruction process, and data analysis programmers for revision purposes (Hansen, 1968). Although this approach is used at many TPI centers (Duke, 1969), many universities entering the TPI development area have avoided the tremendous commitment such an organization implies by offering TPI programming tools, training, and consultation help to instructors who are interested in developing their own course (Christopher, 1970). This approach has the advantage of only requiring a minimal staff but it assumes that there is a sufficient number of instructors willing to invest their time in course development. According to Luskin (1970), however, two of the most critical obstacles to course development are "availability of individuals with appropriate component skills" and "lack of incentives to stimulate the preparation of educational software."1

Another critical obstacle to TPI program development is the high ratio of development time to instruction time. Rogers (1968) reports as high as 300 hours of development for one hour of instruction and Oldehoeft and Conte (1968) estimate approximately 100 hours

1The University of Illinois is considering royalties for course authors (Alpert & Bitzer, 1970).
development time for a 110 minute lesson. To solve this problem some organizations are trying to develop author languages more subject oriented and less computer oriented (Frye, 1968; Hewlett-Packard, 1971; Ting, 1969).

TPI program modification and maintenance is just as important as program development. This activity requires the availability of programs for analyzing student responses and modifying the course. According to Hansen (1968):

The number of revision cycles required to develop an "acceptable version" of a CAI course remains an unanswered question. . . . One would hope that the behavioral evidence might decide the issue, but then the vast majority of CAI learning responses remain unanalyzed due to limited data management systems within the computer. In essence, the revision process within CAI course development will be a major difficulty for any new curriculum project [p. 70].

Fortunately, many of the current course development efforts such as project IMPACT at HumRRO (Kopstein, 1969) and the WSUCAI system at Washington State University (Ting, 1969) have recognized the need for data management programs and have incorporated them into their systems.

Examples of TPI systems. Probably the most successful form of TPI for mathematics has been the arithmetic drill-and-practice programs developed at Stanford University. These programs have been used in Palo Alto, California (Suppes, 1968); McCombs, Mississippi (Carruth, 1970; Prince, 1969); New York City (Abramson and Weiner, 1969; Weiner, 1970); and have been incorporated into Hewlett Packard's education software (Hewlett-Packard, 1970). The Stanford D&P programs present arithmetic drill questions to the student with the level of difficulty of the question being dependent on the student's results on previous lessons. One of the reasons for the wide acceptance of the Stanford programs is
that the sessions are kept very short because of limited branching and a time limit on student response.

Two of the organizations that have conducted large scale tutorial development efforts are the University of Illinois and Florida State University. The Computer-Assisted Instruction Center at Florida State University (FSU) emphasizes the systematic approach to course generation and has generated over 1100 hours of instructional material in 2 1/2 years (Hansen, 1970). Also emphasized at FSU is the use of Computer Managed Instruction to integrate TPI with other course activities. The PLATO system at the University of Illinois has also generated many TPI courses with the main emphasis of this system being to reduce the costs of TPI through computer hardware and software development (Alpert & Bitzer, 1970).

Two more recently developed but less extensive systems are the IMPACT system at HumRRO (Human Resources Research Organization) and the WSUCAI system at Washington State University. The IMPACT effort is not committed to producing a large number of courses but rather to developing a cost effective TPI system. It plans to generate this cost effective system by developing an adaptive model that will "reflect the ideal decision process of an instructor selecting the best action in dealing with any given student at a special time [Kopstein, 1969, p. 26]." The WSUCAI system includes an instructional subsystem, recording subsystem, and management subsystem. The instructional subsystem is designed to allow both tutorial and dialogue modes and provides for course generation capability in English (Ting, 1969).
Two recent efforts in developing multi-media instruction are the PYRAMID system developed by Ampex Corporation and the Southwest Region Educational Computer Network at the University of Texas. The PYRAMID system is a computer-controlled library of taped audio and visual instructional materials which may be accessed from individual carrels at the schools or from home telephones (Johnson, 1971). The Southwest Region Educational Computer Network is experimenting with commercial television sets as terminals. If the technical feasibility of this system can be demonstrated, it will have the capability of incorporating computer generated text with video tapes of classroom lectures (Bunderson, 1971).

Effectiveness. The four variables "time," "cost," "attitude," and "achievement" are normally used by school administrators in choosing from alternative instructional programs with "cost" often being the most dominant. Some administrators are willing to spend additional funds for a new instructional program if it can improve overall student achievement or reduce the amount of student time to learn a given subject area; and, occasionally, equal-cost, equal-effectiveness programs have been differentiated and selected on the basis of improving student attitude. However, the majority of decisions usually reduce to: How much more achievement am I going to get for the extra funds?"

The problem of cost can best be described by a quote from Blair (1968):

In every case, direct instruction by computer has substantial potential; it is effective, flexible and well-received by students and faculty. But every case has also demonstrated that such instruction is not economically viable. Resolving this conflict is the crux of useful CAI.
Gentile (1967) supports this view with the statement "the present costs of CAI are prohibitive for all uses except research." Lynch (1971) concluded, after a cost-effectiveness analysis, that "CAI can be economically justified in institutions of higher education only if the demand for CAI instructional time is extremely large." It is assumed that these authors are not considering the drill-and-practice mode of terminal presented instruction which has already demonstrated commercial viability (Hewlett-Packard, 1971).

Why do the other modes of TPI cost so much? One reason, of course, is the investment in manpower and computer time in generating the course presentations. This expense will be defined as "development costs." The ranges that have been presented for this cost are $400 to $800 per hour of instruction by Alpert and Bitzer (1970), up to $6000 per hour by Hansen (1968), and $200 to $2000 per hour by Zinn (1968). It should be noted that these costs are a function of the sophistication of the presentation strategy and the complexity of the subject matter.

The other major cost factor is the presentation cost. This cost normally includes the use of computer hardware; storage of courses in auxiliary memory (magnetic tape, disks, etc.); communication costs; and terminal costs. Presentation cost is often combined with development cost to determine the cost of a "student contact hour." This cost has been reported to range from $2.63 to $3.48 by Oldehoeft and Conte (1971); from $2 to $5 by Alpert and Bitzer (1970); and from

---

1Does not include developmental and proctoring costs.
$2 to $15 by Zinn (1968) with the last two reports predicting a reduction to the 20¢ to 40¢ range by 1972. This 20¢ to 40¢ projection is based on the use of the PLATO IV system at the University of Illinois with its inexpensive plasma display terminal and its dedicated large scale computer. The validity of this projection has been questioned by Castle (1970) who feels that it underestimates development costs and overestimates the number of terminals that will tie into the system (4000) and the number of hours of usage per terminal (8 hours per day for 300 days).

Regardless of the validity of the PLATO IV estimates, its eventual cost should be considerably lower than most current systems because, through its development of the plasma terminal, it has addressed one of the most dominating components of presentation costs, namely terminal costs. The major problem causing high terminal costs for instruction is the fact that one terminal can only assist 6-10 students during a normal school day if the average session is more than 30 minutes. This implies that to satisfy a school's requirements, the number of terminals would exceed the number of teacher desks, blackboards, etc. which are usually allocated at a 25 to 1 ratio. At an average terminal cost of $100 per month rental, this overall high cost led Ludwig Braun (1970) to state, "The individual use of terminals for CAI is very expensive and cannot even be attempted in most high schools." Mrs. Evelyn Rhinehart of Fairview High School in Dayton, Ohio states that cost of using a short TPI segment offered by General Electric (GE) over the teletype for learning how to use G.E.'s timesharing system is not justified when compared to the amount of student programming that
could be supported for the same cost.

Regarding achievement, a survey of the experimental results of comparisons of TPI to conventional instruction would not be very encouraging for TPI advocates. Abramson (1971) reported no significant differences (NSD) for the second year of the New York City TPI program in elementary mathematics. Baudree (1968) reported NSD between reading and TPI classes for 95 Florida State students. Crawford (1970) reported NSD for seventh grade drill-and-practice. Carruth (1970) reported D&P did not meet the needs of the lowest level students in 4th, 5th, and 6th grades in McCombs, Mississippi. Oldhoert (1970) reported NSD for a numerical methods course at Purdue. Proctor (1968) found NSD for a curriculum concepts course at Florida State University. It should be noted, however, that many of these courses were first attempts, inhibited by overly complicated non-interactive author languages, isolated from other computer-oriented instructional activities, and forced to be presented on teletypewriters instead of CRT's.

There have been some positive results obtained when using TPI. Lorber (1970) reported that the raw mean of the TPI group was significantly higher than a conventional control group for a college course in "Instructional Process and Curriculum." Jamison (1970) reported that in the McCombs, Mississippi D&P program the experimental groups did better than the control groups for all six grades. Alpert and Bitzer (1970) reported that the PLATO group did as well as the control group in a medical science class but spent less time on the course and had greater retention. This change of emphasis from trying to justify TPI on improved achievement alone to justifying it on reduced student
time spent seems consistent with Mitzel's contention (1970) that we are probably not evaluating TPI properly. Particularly, he asks, "Is it possible that CAI offers opportunities to reach cognitive instructional objectives to which users of conventional methods do not aspire?" Alpert and Bitzer (1970) suggest that "a valid measure of the effectiveness of Computer Based Education [another name for TPI] calls for a much larger sampling of data and a longer period of comparison than previously."

**Student and teacher reaction.** Most research has shown positive attitudes toward TPI. Crawford (1970) reports that for seventh grade drill-and-practice in remedial mathematics there was a "marked increase in the experimental (D&P) group's attitudes towards mathematics." Tresize (1970) states

"the fact that CAI programs seem to motivate nearly all youngsters in an extremely important point, possibly because a) they can proceed at their own rate, b) they are fascinated, c) the computer does not insult or chide, and d) it reacts immediately."

Suppes (1968) reported only, 2% of the students taking Stanford's Drill and Practice Arithmetic Program during the 1965-66 school year did not like it.

Regarding the tutorial mode of presentation, Lorber (1970) reports that "students who used CAI were pleased with it as an instructional device, did not find it cold or impersonal, and desired further contact with CAI both as users and authors." Other positive attitudinal results are reported by Weiner (1970), Abramson and Weiner (1971), and Oldehoeft (1970). One negative attitudinal result Ostheimer, 1970) was due primarily to system inflexibility and system malfunction.
One often-used criticism of TPI has been its occasional anxiety-producing effects, particularly when timed drills are used. One of the conclusions of the evaluation of the 1965-66 Stanford D&P program was that "the teachers questioned the appropriateness of the computer-based environment for some children who have special intellectual and emotional problems [Suppes, 1968]." Gerry (1970) reported that although 90 out of 110 college level social science and education students chose TPI to a human tutor, the high anxiety rated students preferred the human tutor. Another criticism of TPI is reduced socialization among students and between students and teachers (Feldman, 1970; Maclaine, 1969).

Implications. To summarize the information presented so far: TPI forces an analysis of the organization of the course material; it is usually well accepted by students and teachers; generation and dissemination of course materials is limited by an insufficient supply of qualified course developers; it costs a lot, with the cost of presentation increasing with the sophistication of the mode of presentation; and, except for a few cases, its effectiveness in increasing learning is questionable at present. These results seem to imply that if TPI is going to be integrated into a course, instructional objectives must be established; and, since development time is so extensive, care should be taken to not include more TPI than could feasibly be developed. One suggestion by Arnett and Duke (1970) seems applicable: "begin with a good, recognized textbook."

With regard to choosing an instructional strategy, the facts that D&P has been successfully implemented on a minicomputer (Hewlett-
Packard, 1970), that a typewriter terminal is recommended for D&P (McConnell, 1968) and that tutorial TPI is too terminal consuming seem to indicate that some form of D&P be used for this study. However, a compromise between tutorial and D&P may be possible by including some textual presentation and branching, while maintaining the short interaction time and restricted responses of D&P.

2.3 Instructional Management Systems (IMS)

Computers are tools to help man think. They do not, strictly speaking, ever make decisions. We make decisions when we write programs which specify what is to be done under each foreseeable circumstance. In so doing, our values are exposed—to our own view and to others. Some of the resistance to the use of computers in the management of human institutions may arise from anxiety about the computer imperative: the need to make explicit the bases upon which we intend to take actions and to select among alternatives. However, properly understood, this imperative may in the long run not only require but permit more creative and flexible management and control systems in education with the object of improving our capacity to predict consequences and to respond to rapidly changing circumstances. (Caffrey, 1965, p. 1)

The major functions of an instructional management system (IMS) are the collection of student and resource data, summarizing this data into useful reports, and associating this data with instructional objectives to prescribe student activities. Many of the initial efforts in this area were concerned with just the collection and reporting of student data, i.e., automated testing. These efforts have been relatively successful and have made major impacts in the methodology of testing (Caffrey, 1965).

However, the requirement of specifying instruction objectives for TPI, and other individualized instruction strategies, has resulted in a vast quantity of required record keeping, including records of each objective attained by each student, questions used to test the
attainment of objectives, etc. In the TPI systems it was just a natural extension to include these records in the computer since the TPI programs and data were already there. However, in the non-computerized individualized instruction courses, the record-keeping burden on the teachers has also led to much of this data being computerized.

Although there exist a few comprehensive instructional management systems, including the linking of CMI (Computer Managed Instruction) to TPI at Florida State University, much of the theory and methodology of these systems is as yet undeveloped. Still to be perfected are procedures for evaluating the usefulness of the information in the system, for determining and evaluating prescription algorithms, and, according to Baker (1971), for describing how the teachers employ the computer reports in the management of their classrooms. The remainder of this section will present more details on IMS development and implementation; its organization will be the same as the preceding section's.

**Variables influencing IMS development.** Some of the major variables influencing the design of an IMS are 1) the items in the data base, 2) the currency of the data, 3) the form of input, 4) the computer used, 5) the form of output, and 6) the construction and maintenance of the system. The three major sets of data to be considered for the data base are student records, instructional resources, and the hierarchy of instructional objectives. Student records could include data such as test scores, homework problems done, time spent on homework, experiments completed, etc. (data normally kept in a teachers' gradebook), as well as a record of instructional objectives attained. Resource data could include textbook pages, TPI questions, homework,
etc. related to instructional objectives and could even include complete tests and test-grading keys.

In selecting items for the data base, care must be taken not to specify the items to such a detail that excessive computer memory is required or that the collection process is too costly or time consuming. An example of the type of questions asked during this phase is, "Should the student's answers on individual test items be kept or only his tabulated test scores?"

The currency problem involves defining data storage, collection, and reporting procedures to allow the use of accurate data (to the degree necessary to make correct decisions) at the time needed. Many systems employ daily updates either by summarizing TPI responses collected during the day or by processing a deck of cards turned in after school. Although daily updates are sufficient for many activities, Coulson (1970) recommends that students be assessed and reassigned 20 to 30 times a day. "Currency" also implies the capability to periodically alter the instructional resources data to eliminate ineffective activities and to reassign other activities.

The specification of the input media usually involves selecting or designing a card or form that will minimize both the amount of recording errors and the amount of manual processing required to enter the data into the computer. Because the use of cards (or forms that are used to convert data to cards) requires the availability of a card reader, some IMS systems use the teletypewriter for almost all of their input.

The computer used in an IMS system can dictate the quantity and
quality of data stored, the form of input and output, and the intricacy of the prescription algorithms. Frye (1970) states that "computer-managed instruction and computerized instruction usually require at least a medium-scale machine because of the additional workload it will have." Ferguson (1970) disagrees:

"Experiences which have been the product of the initial implementation of a computer management system for Individualized Prescribed Instruction (IPI),...have resulted in the evaluation of the hypothesis that a small computer could support educational management, computer testing, and some CAI lessons.

Baker (1971) states that "continued reliance on large scale computer installations would seriously hinder development of CBIM systems."

Probably the most crucial activity in assuring the acceptance of the instructional management system by the students and teachers is the designing of useful querying commands and attractive, concise standardized reports. This implies that the users of the reports, whether students or teachers, should be involved in their specification. Some of the options available in designing reports include the capability of generating summary reports and/or reports of exceptional behavior. Another consideration is the form and frequency of prescriptive reports. Although the data base and input/output forms are specified by educators, the construction and maintenance of the IMS is usually accomplished by a team of computer professionals. There has been considerable progress recently in developing generalized data management systems, but these new systems will still probably require the services of computer professionals. However, they do offer potential cost and time savings in IMS development.

1Computer Based Instructional Management
Example of IMS systems. The University of Pittsburgh's Research and Development Center recently computerized their IPI Individualized Prescribed Instruction) program (Sass, 1970). The student input into this system is pencil-marked sheets which are converted to punched cards via an optical scanner, normally a two day process. The punch cards are then read into the computer through a remote batch terminal located at the school. The reports generated by the system include unit summaries for each student, homeroom summaries, and overall unit summaries. A similar system at the University of Wisconsin, Individualized Mathematics Curriculum Project (IMCP), collects responses to multiple-choice tests to determine if a student achieved a given objective (Baker, 1971). Most of the reporting is via a teletype and is initiated by teacher inquiries.

Westinghouse Learning Corporation offers a commercial service called PLAN (Program for Learning in Accordance with Needs) which automatically administers a course according to teacher-supplied input. This input includes:

- a comprehensive set of educational objectives
- alternative means of meeting objectives
- criterion referenced tests related to objectives
- guidance and individual planning procedures

Student input to this system is on-line via a teletype (Baker, 1971).

Systems Development Corporation (SDC) in cooperation with the Southwest Regional Laboratory has experimented with instructional management systems for first grade math and reading (Bratten, 1968). Thirty-five tests were administered throughout the reading course for
four classrooms at two different schools. Input was via an optical scanner at SDC's computer facility. Including math and reading, more than 500 separate performance records were kept for each student taking the course. Remedial activities were prescribed if the overall group level was below 85%. The reports were used by the teachers to make instructional decisions about 1) pacing the class, 2) reassignment of students to different ability groups, 3) administration of supplementary remedial materials, 4) modifying the sequence of instruction, and 5) revising the instructional objectives. The teachers, not being completely satisfied with the daily reports supplied by SDC, asked for reports that were less detailed and more comprehensive.

One of the more sophisticated reporting schemes belongs to the PACER (Prescriptive Analysis of Curriculum Evaluation) system at Oakland, Michigan Schools (Joos, 1970). This system has four levels of reports: 1) a listing of the concepts that each student may have attained, 2) a listing of the number of students of each achievement level that choose each answer to a particular test item, 3) a summarization of results for each achievement group of each class for each school in the school district, and 4) a summarization of results for the combined school district.

Effectiveness. The development of instructional management systems requires much coordination between user and developer with many important modifications taking place as the user becomes more familiar with the information furnished him. Although this evolution is necessary for effective systems, it makes comparisons with other instructional methods very difficult. However, despite their evolutionary nature,
some computer managed courses have shown improved achievement in comparative evaluations.

Forbes (1971) reports that for a college course in introductory mathematics, the grading and reporting of weekly quizzes and homework kept the students from getting behind, had fewer failures, and "resulted in substantial improvement in student performance." In a similar course at Winona State College, Underkoffler (1969) reports that the experimental group whose weekly homework was scored by computer, with results and prescriptions immediately available, achieved significantly better than the control group whose homework results were reported the next class session. Other achievement improvements have been reported by Daniel Miller (1970), Prince (1969), and Paden (1970).

However, not all results have been positive. Silberman (1968) reports no significant difference in achievement in first grade reading using the SDC Instructional Management System. He suggests that this lack of significant improvement over traditional teaching may have been due to developmental imperfections, prescriptions not always followed by the teachers, imperfect diagnoses, ineffective supplemental materials, and the slow cycle time in recognizing ineffective remedial prescriptions. Sass (1970) reported that for the computerized version of the IPI (Individualized Prescribed Instruction) program at Oakleaf School in Pittsburgh, the students were not mature enough to select from the options suggested in the computer printout.

In the area of testing, considerable success has been reported by Ferguson (1970), Woods (1969), and Caffrey (1965) in using terminal
presented diagnostic tests to reduce test-taking time and to increase the predictive accuracy of the testing process.

Attitudes. Instructional management systems have usually been well accepted by the teachers and students involved. The teachers using the SDC first grade reading and arithmetic program reported that the children enjoyed working with the Instructional Management System, that planning required less time, and that it was easier to explain pupil progress to parents (Silberman, 1968). Baker (1971) reported that 86% of the students using the TIPS (Teaching Information Processing System) during economics courses at the University of Wisconsin "felt that the system helped them." Joos (1970), using the PACER program at Oakland, Michigan schools, reported that "the teachers do not find...detailed analysis threatening as long as the listings are carefully explained and widely used." In addition, he reported that most of the teachers agreed that the reports provided "real truth and introspection."

Probably the most meaningful demonstration of the motivational value of instructional management systems, as far as the author is concerned, is his own experiences in using the computer to manage student data for a computer concepts course at Wright-Patterson Air Force Base. This course uses the computer each session for quiz grading and providing individual student and teacher reports on attendance, quiz records, homework records, and time spent on homework. These procedures eliminate the time spent in correcting quizzes for a class of 54 students and allow identification of students having difficulty, poor quiz questions, and unreasonable or ineffective
homework assignments. However, the most startling observation has been the student's reaction to their individual reports. Many of the more than 200 students who have taken the course have shown concern when there are slight errors in their student records and have expressed disappointment when their individual reports were inadvertently delayed. It should be noted that this is a non-credit course with students ranging in military rank from Airman to Colonel and in education from high school to Ph.D.

**Implications.** To summarize the information presented so far: IMS has been most effective in the testing environment; reporting student and resource data has been well-accepted by students and teachers and has allowed them to "manage themselves"; and the development of IMS systems requires coordination among computer experts, educators, and the eventual users of the system. These results seem to indicate that if a suitable media and input device is available, computer scoring of classroom administered quizzes is both feasible and effective, especially if useful reports are also generated. The discussion on system design emphasizes the need to specify instructional objectives, arrange them in a hierarchy, and associate them with instruction resources.

The choice of a computer to implement an IMS seems to indicate a need to try one on a minicomputer but implies that a computer expert will still be needed. The lack of individual prescription in some of the systems surveyed seems to indicate that the development and implementation of prescriptive procedures is no simple task and, therefore, implies that any initial efforts should strive for simplicity. An
overall impression of the development of IMS systems is that many revision cycles are necessary before a truly efficient system is available.

2.4 Mathematics Information Systems

As a consequence of the explosion of mathematical papers and results in the last few decades, a situation has been reached in which experts even in related fields of mathematics speak a different language, in which no mathematician can read a mathematical journal from cover to cover, and in which it is becoming increasingly difficult for one to keep abreast of one's own field, not to mention related fields. (Pager, 1972, p. 71)

As a solution to this mathematics information explosion, which is just a part of a larger scientific information explosion (SATCOM, 1969), Pager proposes a "multirooted tree of all mathematical theorems and terminology to be maintained in an interactive computer system [p. 72]." Although Pager is addressing the complex problem of organizing all of mathematics research, might not a first attempt at such a system be to limit its content to secondary school mathematics? Figure 1 represents a simulated dialogue with such a system.

Many of the proponents of "dialogue" TPI and/or student-initiated learning have tacitly assumed the existence of an information system in specific education areas. Nelson (1970) recommends the existence of such a system that cuts across traditional curriculum boundaries. Berkeley (1971) calls the combination of an information system with a program for monitoring student interaction "Learner-Controlled CAI." He states that in such a system "curiosity will motivate [and] learning will become goal-oriented." In support of student-initiated learning, the Commission on Education of the National Academy of Engineering (1969) states that "training a university student to be a
Fig. 1 Simulated dialogue with a mathematics information system
competent self-learner is the most important and lasting contribution a school can make." However, if one assumes that student use of an information system fosters self-learning traits, then the critical issues seem to be: 1) "At what age or maturity level are the students able to assume the responsibilities such a system requires?" and 2) "How can the use of an information system be made attractive to the student?"

The remainder of this section will discuss some of the considerations in developing an information system and describe a few examples of these systems in secondary education. It will conclude with a discussion of the implications of these systems for this study and for secondary mathematics in general.

**Developmental considerations.** Some of the questions asked during the design of an information system are:

- How should the information be stored in the computer? Keywords? Bibliographic form? Abstracts? Complete Documents?
- What is the most inexpensive procedure for getting data into the system?
- How should the information be arranged once it is in the computer?
- Should the system be interactive?
- If interactive, how complicated can the queries be?
- Should the information system be able to communicate with other systems within the computer?

Most of these questions are resolved by information scientists after surveying the informational needs and backgrounds of the intended audiences and ascertaining the capabilities of the computer resources available.
Examples of mathematics information systems. Wexler (1970) describes a system at the University of Wisconsin which contains subject matter in geography, biology, vocabulary, and some physical sciences. This information is available to teachers when generating TPI sequences using TPL (Teacher Programming Language), to students for remediation while being tutored at the terminal, or in the dialogue mode during which either the student or the teacher can browse through portions of the information net extracting selected information. This system is administered by a large scale computer, a Burroughs B5500. Marks (1970) describes a similar system called CAI MARK at Iowa State University. The CAIMARK system combines a computer-based problem-solving environment with an information file and allows the student complete control of all components.

An innovative system called Project Xanadu is under development by the Nelson Organization (Nelson, 1970). This system is designed to free students from conventional teaching techniques by allowing them control of an interactive information system which presents a variety of interesting displays, called Hypermedia, on a video terminal. Hypermedia are further classified as Hypergrams, Stretch Texts, Hypermaps, and Hypercomics. An example of a Hypergram is a student dissecting a frog by using a light pen against an image of a frog displayed in color on a CRT. A Stretch Text allows a student to expand a text to include more explanations or definitions if the original text is too difficult to comprehend. A Hypermap blows up selected portions of a map displayed on a CRT and presents geographic, economic, and cultural information if requested. A Hypercomic displays comic
characters discussing concepts with different type characters explaining the text in different styles of conversation (technical, conversational, slang, etc.).

Effectiveness. The author was able to find only one comparative evaluation of the use of an educational informative system. This experiment (Barnes, 1970) compared "learner-controlled CAI" to "conventional CAI" and found no significant difference in achievement. Barnes concluded that

if learner-controlled CAI can significantly affect performance, it must have the following three prerequisites:
1) the learner must be ready to assume responsibilities,
2) the material must be meaningful and relevant to the needs of the student, and
3) the learner must be motivated to learn the material.

Implications. Mathematics information systems seem to offer both a convenient source of reference and a stimulus to further study within and across subject areas. An additional benefit that may be even more valuable than the student initiated learning is the forced analysis of mathematical definitions, symbols, theorem dependencies, etc. required to precisely define the system. Unfortunately, storing of large quantities of textual data in a computer is still quite expensive and therefore precludes the use of this activity for the majority of secondary schools. However, Berkeley (1971) predicts that the costs of these systems will decrease as more minicomputers enter the schools. Nelson (1970) predicts a substantial decrease in terminal costs for these systems by 1975 and emphasizes that Project Xanadu will have all of its functions, including the Hypermedia, handled by a minicomputer with tape and disks as auxiliary storage.
Although both the above authors predicted minicomputers will be used for implementing information systems in secondary schools, the HP 2114B minicomputer to be used for this study has the capacity of storing at most 4000 characters; this is less than two pages of the textbook to be used for the course. Consequently, mathematics information systems will not be considered for this study.

2.5 Computation Tool

Although the differences between pure computation and evaluating an algorithm were explained in Section 1.4, a further example might be helpful. Suppose a student needed to know the square root of 33.9 in order to answer a TPI question. Since he is already connected to a computer, but as part of a different application, the following simulated instructions could possibly solve his problem:

```plaintext
SAVE TPI
BASIC
10 PRINT SQR(33.9)
RUN
5.82237
RETURN TPI
```

This sequence of instructions only performs a calculation and does not evaluate a algorithm. Many time-sharing systems have the above capability which can be used in conjunction with TPI, programming, or information retrieval. This capability is often called the "calculator mode." If the language BASIC is part of the available software, it is often used in this mode because of the versatility of its PRINT statement.

Because computation is normally included in other computer applications, very little has been reported about pure computational activities; however one interesting application, reported by Pavlovich
(1972), is the use of the computer to calculate and list many terms in a sequence that had failed all of the easy-to-use analytic convergence tests. Although the computer results could not be used to prove or disprove convergency, they did present evidence regarding the final outcome and helped eliminate many futile approaches.

Implications. The use of preprogrammed procedures, often called "canned subroutines," has allowed scientific researchers (and high school science students) to concentrate more on the results of their experiments and less on computational procedures. The use of preprogrammed mathematics procedures offers similar advantages to secondary school mathematical problem solving by allowing complicated problems such as matrix applications, linear programming, statistical analysis, differential equations, etc. to be considered.

With regards to this study, the student programming effort could be reduced by offering preprogrammed geometric procedures for finding distances, equations of lines, etc. However, since the students should also know how to perform these calculations, the approach taken will be to require them to build their own subroutine library as they progress from the straightforward to the more complex problems.
2.6 Evaluation of Algorithms

Although much of the theory of algorithms and computation was developed by mathematicians and logicians such as Church, Post, Gödel, Turing, and Markov during the 1930's, little attention was paid to this problem solving technique in secondary mathematics curriculums of the forties and fifties because of the time-consuming process of carrying algorithms to completion. However, when the processing of algorithms did become feasible because of computers, a few startling observations were made by mathematicians and mathematics educators. One of these was the increased availability of constructive existence proofs which are always more desirable than non-constructive proofs. Another observation was that algorithmizing the solution to one particular problem forces the problem solver to consider the total class of problems for which his problem belongs. This awareness of the large class of problems doesn't necessarily require the problem solver to algorithmize the total class or problems but does require him to account for the particular assumptions and characteristics of the subclass of programs he is trying to solve.

A third observation was that a whole new world of non-analytical mathematics such as finding solutions of transcendental equations, finding roots of polynomials, and performing matrix manipulations was now available to the students with non-trivial examples and problems. The Committee on the Undergraduate Programs in Mathematics (1965) states that

The prevalence of the high speed automatic computer affects the teaching of mathematics in a very general way. Many mathematically trained students will work closely with computers, and even those who do not, should be taught to appreciate the type of al-
algorithmic approach that enables a problem to be handled by a machine. This point of view should be presented along with the more classical one, at appropriate places in calculus, differential equations, linear algebra, etc.

Finally, because the computer was such a "perfect" executor of the algorithms, it was observed that the problem solver was forced to write a "perfect" algorithm. Morton (1970) states that one of the major advantages of computer problem solving is that "The student must thoroughly understand every problem he submits to the computer for solution."

Many mathematics educators have recognized the value of algorithms and have incorporated them into their mathematics courses. In fact, this activity is probably the most popular in primary and secondary schools, not only because of its educational advantages but because it requires little hardware and software support, minimal teacher-computer experience, and is relatively inexpensive. Although there have been many instances of creative students solving complicated problems beyond the scope of their current courses, the majority of applications have been within the framework of traditional courses. Although a few textbook series have been written to assist the integration of this activity, the proper choice and number of problems for a given course is still unresolved.

In comparative evaluations, the results have been rather mixed although this activity appears to be most effective for the higher-ability level students in the lower grades and lower-ability students in the upper grades and college. Attitudes have also been mixed with some instances of students becoming disenchanted with this activity after initial enthusiasm (Hahn, 1970). The remainder of this section
will discuss in more detail some of the variables to be considered in implementing this activity: some of the current large-scale efforts, and some of the comparative results regarding effectiveness and attitude; it will conclude with some implications for this study.

Variables. A review of the literature suggests that the major variables influencing the use of the computer for evaluating algorithms are 1) grade level and topic, 2) language, 3) computer interaction, and 4) number and quality of programs. Regarding grade level, programming has been taught to first graders (Austin, 1969), sixth graders (Computers and Automation, 1971), and in junior high school (Altoona, 1968; Harvey, 1968; Hatfield, 1969; Washburn, 1969). At the high school level, computer evaluation of algorithms has been tried in almost every topic with algebra being the most common. In college, calculus has been the course most commonly tried, with differential equations and linear algebra also being considered. One participant at the 1970 CRICISAM conference stated that "all theory of linear algebra can be presented from an algorithmic point of view [CRICISAM, 1970]."

The two most commonly used languages to program algorithms are BASIC and Fortran. In Altoona, Pennsylvania schools, BASIC is taught to eighth and ninth graders and Fortran to tenth through twelfth graders. According to Frye (1970):

The strength of BASIC is its simplicity. It has been used successfully in junior high school, and only rudimentary instruction is needed to prepare a student to use it - students can be working on their own with BASIC after as little as two hours of instruction.

---

1Center for Research in College Instruction in Science and Mathematics located at Florida State University.
He adds: "BASIC is the most widely available of the problem solving languages." The main difference between BASIC and Fortran is the lack of input and output formatting requirements in BASIC and its limited ability to use subprograms. These differences normally favor BASIC for short programs with little input-output, typical of the programs used to evaluate algorithms in mathematics courses. Fortran is usually recommended for complicated programs with many subprograms and is occasionally chosen because it is the language most commonly used for scientific problem solving in industry.

Concerning computer interaction, limited terminal availability has forced many schools to adopt a "pseudo-batch" mode of input for student programs. The use of inexpensive off-line teletypes, and optical card readers that can read programs marked on cards with an ordinary pencil, has considerably increased the number of students that can use the computer during a given day. This off-line program preparation and debugging has not influenced the quality of the programs generated and, in fact, may be more effective. According to Katz (1971), "the most effective method of computer utilization appeared to be program writing with no direct computer access, as opposed to direct access." Some schools still use keypunches to generate card decks to be run at minimal cost on computers at local industries or colleges, but this procedure is becoming less popular due to transportation problems and poor turnaround.

---

1 About half the cost of on-line teletypes.
The proper number of programs to be written for a course is a function of course content, available facilities, and students programming background. For a time-sharing environment, The Dartmouth Secondary School Project (Nevison, 1970) recommends "three [programs] a week in public schools, perhaps four a week in private schools [p. 17]." In the card-oriented batch processing mode with one day turnaround, most courses cannot require more than five to ten programs of any degree of difficulty. The average number of programs reported for the first semester of the CRICISAM calculus course was 1² with 10 being the most common amount; one instructor assigned 25 programs but felt that it might have been too much (CRICISAM, 1970).

Although it is desirable to check the quality of programs to identify poorly learned mathematics concepts, the usual criteria of evaluation is whether the program generates correct answers. One method of determining correctness used in the SIMON (Simple Instructional Monitor) system is to compare the students program to a "true" program (Peurzeg, 1970).

Large-scale efforts. The Dartmouth Secondary School Project was conducted from 1967 to 1970 with an emphasis on letting students and teachers "explore together the classroom and extracurricular uses of computing [Nevison, 1970, p. 5]." More than 3000 students from 25 participating secondary schools used the Dartmouth Time Sharing System with BASIC as their principal language. The conclusions after the first year of the study were:

- The computer is best used to explore problems of personal interest to the user, i.e., as a creative outlet for his curiosity.
Programming courses are obsolete. Developing the ability to program is a matter of a few hours.

Teletype time is too precious to be wasted by using the computer as a teaching machine.

The average student can learn to program in seventh grade.

The major influence a student's ability has on using the computer is the length of time required to do a task. Even very slow students can productively use a computer [p.5].

A different approach has been taken by the CRICISAM and CAMP (Computer Assisted Mathematics Program) projects. These efforts are more oriented toward incorporating the use of algorithms into traditional courses. The CRICISAM project has generated a set of textbooks for computer-oriented calculus which are being used at 50 universities (Miles, 1971). The CAMP project is more oriented toward junior high and high school mathematics and has generated a series of computer oriented textbooks for many of the courses at this level (Johnson, 1968). Constructing and evaluating algorithms is the major computer activity of both of these programs.

Effectiveness. The effectiveness of this activity will be discussed separately for calculus, algebra, and junior high school mathematics. In an experiment with calculus, Bell (1970) reports that

The conclusions of the experiment support the hypothesis that a computer-oriented approach to calculus is an effective method to promote understanding of concepts and to increase student interest in calculus, and it does not interfere with the students' learning to apply the techniques of calculus.

Bitter (1970) reports that for a college introductory calculus course using BASIC, the experimental group achieved significantly higher on differential calculus although there was no significant difference on integral calculus test items. Holoien (1970) concluded that "a
computer seemed helpful in learning calculus, probably more so for lower-ability college students than for higher-ability ones." A non-positive result was reported by Fielder (1969) for a course in analytic geometry and calculus. The experimental students programmed their homework in Fortran while the control group did pencil and paper homework. There was no significant difference in scores on four instruments containing 35 multiple-choice items each.

Considering the topic of algebra, Altoona Public Schools (1968) reported that "the experimental group using BASIC attained a higher level of learning in algebraic problem-solving, especially arithmetic reasoning." Similar positive results for algebra courses are reported by Kieren (1969) and Ronan (1971). A negative result in an algebra course, of particular significance to this study, is reported by Katz (1971). He used programming on a PDP-8/s minicomputer to supplement a conventional course and attributed the lack of achievement of the experimental group to the extra time spent in the computer room waiting to run programs.

In junior high school mathematics, Hatfield (1969) concluded that his results should lend support to computer-assisted problem solving in grade 7 mathematics particularly for study of number theory, for ability to solve unfamiliar word problems, and for problem solving in general [when used by] students identified as "high" and "average" achievers.

Washburn (1969) experimented with junior high students, twelfth grade students, and college freshman. The experimental classes wrote computer programs in lieu of homework. He concluded:

the writing, execution, and correction of computer programs can
strengthen one's understanding of mathematical concepts. This
gain is independent of one's age and level of mathematics ac-
achievement. Although students of higher intelligence tend to
derive greater benefit, students of average or lower intelli-
gence benefit as well.

Foster and Johnson (1972) reported positive results when using the
computer and flow charts for eighth grade problem solving.

Attitudes. Attitudes towards the generation of algorithms and their
evaluation by computers have been mixed. Holoein (1970) reports no
significant difference in attitudes for a college calculus course re-
quiring computer programs for 50% of the homework. Washburn (1969)
reports that

not only is there a strong positive attitude toward the Computer
Enriched Math Program approach [requiring programs for homework],
there is an indication that this approach can also improve stu-
dents' attitudes toward mathematics in general. These attitudes
are independent of one's age, level of mathematics achievement
and intelligence.

At the CRICISAM conference (1970), it was agreed by all that "there
are people who are turned off by the computer." However, only four
participants out of 40 or more attending felt a "hostile-environment"
on their campuses for computer-related calculus and only four partici-
pants noted larger than normal attrition whereas ten noted less than
normal attrition.

Implications. The information presented so far seems to imply that
BASIC is a viable language to use for this activity and that few stu-
dents should have trouble learning and using it. The number of pro-
grams feasible for a secondary course seems to be much less than that
planned for this study; however, the programs for this study may not
be as difficult as those used in the reported efforts. Little was
reported regarding the use of check programs, but this was probably
due to most of the courses not requiring very many programs.

2.7 Demonstration of Mathematical Concepts or Principles.

One of the principal complaints of the use of the computer, or
terminals, in the classroom is that it is too expensive a device to be
dominated by a few students. Braun (1970) states that "the individ-
dualized use of terminals for CAI is very expensive and cannot even be
attempted in most high schools." He then recommends that teachers be
encouraged "to use computers as they would use audiovisual equipment."
The application of the computer to be discussed in this section could
possibly be called "using the computer as a audiovisual aid" except
that the computer seems to offer so much more versatility for class-
room demonstration than conventional AV equipment.

The demonstrations to be discussed in the section can be classi-
ified as "active" (simulation and investigative programs) and "passive"
(graphic displays and animated movies). In the active mode, simula-
tion has received widespread acceptance in science and economics
courses when suitable terminals and software are available; however,
its use in the mathematics curriculum has been limited mostly to
courses in probability and statistics. Investigative programs have
been used much more extensively in mathematics, particularly for esti-
mating roots of equations and limits of sequences.

In the passive mode, the use of the computer for generating
graphics displays and organizing their presentation is still in its
infancy. Particularly, the hardware and software required for these
demonstrations is still too expensive for secondary school use.
However, the many recent technological developments in this area should very shortly eliminate these cost constraints. The potential of this activity is well described by Siklossy (1971): "The blackboard does not lend itself to dynamic manipulations compared to graphic display or CRT." However, the proper use of graphic displays will be contingent upon their acceptance by teachers, and, if accepted, upon effective training in the methodology of their use. This methodology is far from established and much research needs to be done before such teacher training can become part of teacher education curricula.

Very little has been reported on the effectiveness of computerized demonstration techniques either because there are few comparative alternatives, as in the case of simulation, or because the demonstrations are included as components of larger computer-based experiments. However if the cliche "a picture is worth a thousand words" has any validity, and if there exist mathematics teachers who draw poorly, then the automated drawing of complicated diagrams should be effective if for no other reasons than improving teacher morale and reducing unproductive class time used to make drawings. The remainder of this section will present more detail about these systems and will be organized in the same manner as the previous sections.

Variables. A review of the literature suggests that the variables influencing the development of the applications are 1) mode (active or passive), 2) subject, 3) demonstration device, and 4) generation of supportive programs. Since "mode" has already been described above, this discussion will begin with the choice of subject.
Graphics demonstrations have been used effectively in analytical geometry and numerical methods courses, and simulations have been used in probability and statistics courses to demonstrate stochastic behavior. Investigative programs have been used in almost every mathematics course. Although simulation has not been extensively used, its interactive nature and its requirement of making accountable decisions seems to offer great potential for many courses by increasing student activity and by eliminating the artificiality of word programs.

Many improvements have been made in the available display devices for this application. There are now colored CRT's and 3-dimensional CRT's, on-line plotters for graphic curves and charts, overhead displays for teletype terminals, printer terminals that also plot to a .05 inch resolution (Typagraph, 1970), and TV screens attached to programmable calculators (Hewlett-Packard, 1969). In addition, the computer can offer animated movies of geometrical concepts by photographing incremental plots from a program designed to systematically change features of the display to cause the effect of motion. For permanent recording of a graphics display, computer-output microfilm (COM) can be used to save the original display and to generate copies for student distribution. As mentioned previously, most of these devices are too expensive to use now but their use should be feasible in the near future. One very important selection criterion, if demonstrations are to be presented as planned, is the reliability of the terminal and its background computer.
Although there have been instances of teachers and students writing their own simulations, this activity usually requires considerable more programming skill and development time than is available to most mathematics teachers. The generation of displays requires even more programming competence than simulations. The best procedure for implementing this activity, therefore, seems to be for schools, universities, computer manufacturers, software firms, and time-sharing vendors to coordinate their development efforts and to share their supportive software.

Examples. Brook (1970) describes four applications in computer graphics including a course in numerical methods with a graphics display unit used for classroom demonstration of approximation, interpolation, roots of equations, and differential equations. Dwyer (1971) describes a computerized analysis of a bouncing ball with the output on a Hewlett-Packard plotter.

Hicks (1971) describes a system called CIDAC (Communication among Inexpensive Displays And Computer) which emphasizes improved lectures and/or individual problem solving by having available a library of graphical diagrams that can be either plotted at the classroom terminal or displayed on a screen. The diagrams are placed into the system by the lecturer hand-drawing them on a Victor Electrowriter Tablet. They are then plotted or displayed on request with the retrieval and communications being controlled by the computer.

In the simulation mode, Digital Equipment Corporation (DEC) has available, as software on its Edu 20 system, programs to generate Civil War battles, economic simulations, the ecological life cycle
of a pond, and the spread of diseases into epidemics (DEC, 1971). In
the investigative mode, Dennis (1968) describes a geometry course for
junior high school students in which the students used interactive TV
screens to draw polygonal figures and vary these figures against
author specified characteristics.

Effectiveness. The few results that have been reported are encourag-
ing. Oliver (1969) reported increased effectiveness in final exam
scores and a few subunit tests for a course in numerical analysis
taught with the assistance of an IBM 2250 Display Unit. He also made
the qualitative observations that the experimental group had improved
retention, greater class participation, and an intuitive understand-
ing of the applicability and properties of techniques used in numeri-
cal analysis. Dennis (1968) concluded for the teaching of geometric
concepts to junior high students via TV interaction that "the students
were able to gain a verbal knowledge of a large number of properties
of triangles and quadrilaterals" and that they were "also able to ac-
quire a good knowledge of the necessary and sufficient conditions for
the various concepts presented."

Implications. Computer generated displays will not be used during
this study because display hardware will not be available. Although
simulations and investigative programs could be integrated into the
course to be presented, the time to generate these programs would be
more than the author could afford because of the other developmental
efforts. In addition, the use of these programs would decrease the
amount of terminal time available for the other activities.
2.8 Computer as a Subject of Instruction

A popular recent addition to American newspapers has been the column which responds to reader's requests for assistance in solving difficult problems. Many of these problems are concerned with shipment delays, refusal to pay legitimate refunds, and erroneous billings. These problems are usually resolved by the offending company making an apology and blaming the mistake on a conversion to computers or a computer error. Seldom is it mentioned that the errors were really made by the people of the department using the computer and not the computer itself. Although this shift of responsibility seems harmless and offers a "scapegoat" that can't defend itself, it unfortunately causes the perpetuation of the illusion that computers are human and have human characteristics. At the funeral mass for a migrant worker who shot two policemen and then himself when they came to repossess his housetrailer, the presiding bishop criticized our "computerized society" for causing a normally rational, sane person to commit such an atrocity just because of poor business acumen.

Is the computer unjustly blamed for causing many of the ills of our society, or could it be that computer capability has far outdistanced man's ability to adapt this capability to his environment and life styles? In a feature article on computers in the November, 1970 issue of National Geographic, Alan F. Weston of Columbia University is quoted as saying,

Man has progressed over the centuries from the status of a subject of a ruler to that of a citizen in a constitutional state. We must be careful to avert a situation in which the press of government for systematic information and the powerful technology of computers reverse this historical process in the second
half of the 20th century making us "subjects" again [p. 631].

In the same article: U. S. Senator William Proxmire of Wisconsin warns that

as credit-bureau files on some 120 million Americans are computerized, and linked into nationwide data banks, questionable information and data-processing errors are traveling faster and farther than ever.

You could lose your credit, your insurance, even your job, because of such an error in a credit-bureau file [p. 630].

The author of the article, Peter White, asks

How could you correct inaccurate information about you once it was in the government data banks? As things stand now, you couldn't. An error in a personnel file, put in from some extraneous tape, could cost a civil servant a promotion. Or keep a man from getting a job. Chances are he would never know why [p. 630].

One possible solution to improper use of data banks is to pass legislation to protect the individuals' right to privacy, or, at least, to correct personal data. If such legislation is to be encouraged, evaluated, and eventually enacted however, the supporting public must be able to understand the capabilities and limitations of computers. A recent committee of past officers and members of the Association of Computing Machinery (ACM), the professional society of the computing community, has made the following recommendation:

At this time the consensus is that ACM should promote better education for both members and the general public. The public needs to become more aware of the impact of computers on society and how and why most problems are caused not by computers, but by people who program and use computers [Dodd, 1971].

A 1970 survey of secondary schools by the American Institute for Research under contract to the National Science Foundation estimates that 34% of secondary schools have access to and/or use computers. These schools primarily use the computer for teaching computer
science or as a tool in problem solving (Molnar, 1971). The above statistic seems very encouraging until one probes a little deeper into the amount of utilization and realizes that only a very small percentage of students of these schools are actually using and studying the computer. This small percentage is due to the lack of availability of equipment and qualified teachers, with mathematics teachers usually assuming the responsibility for teaching about computers.

However, if the prediction by Post (1970) that "the computer will be a fact of the students' life in years to come" has any validity, then it seems that knowledge of computers should be available to all students at their level of comprehension.

Assuming that courses with the computer as a subject do become available for all students, there is considerable difference of opinion regarding the content of these courses. One view is expressed by John Kemeny, president of Dartmouth and co-author of BASIC: "Our freshman start writing programs after two one-hour lectures. We don't teach them about computers, we teach habits of inquiry [in White, 1970, p. 633]." An opposing view, which relegates programming to a supportive role, emphasizes the social perspective of the computer, its basic concepts and design, and the types and methods of problem solving (Bingham, 1970).

**Implications.** Regardless of whether a separate computer science course is available, mathematics courses that use the computer in a variety of ways can also be effective in teaching its limitations and capabilities. Hopefully, some residual computer knowledge will be obtained by the students participating in this study.
2.9 Integrated Systems

Although many schools have computers, the financial plight of education in general places a responsibility on the school's computer teacher or department to use the machine most effectively. Bell and Moon (1969) suggest that

the high cost of tutorial CAI may be offset by other uses of computers such as:

a. teacher controlled CAI [classroom use of CAI augmented by a TV display,

b. games in math classes,

c. demonstrations in business education, science and statistics, and

d. classroom use of information retrieval.

Lippert (1971) recommends that "computer support of instruction should not be thought of as only teaching programming of offering statistical analysis, or course grading, etc. All are important but not sufficient [p. 41]." He then continues by listing the kinds of activities that would appear to be computer support of instruction and student services:

1. Provide computational time sharing

2. Provide remote job entry in classrooms, laboratories, offices, and study areas

3. Provide programming and software support for teaching faculty

4. Provide computerized academic and career placement counseling

5. Provide user oriented statistical analysis library or applications package

6. Providing Computer Managed Instruction capability
7. Providing CAI, especially to reduce initial course differences

8. Computer based test question library

9. Providing for the creation and availability of two new types of data records — student data record and curricular data record.

Although Nelson (1970) states that "the variety of alternative systems for computer teaching have not even begun to be explored," some efforts are being made in this direction. Dwyer (1971) states that the primary aim of PROJECT SOLO [an experiment in regional computing for secondary school systems in the Pittsburgh area] is not to train students as computer professionals (although they do become amazingly expert) but to explore the full potential of interactive time-sharing computer systems as a learning tool within modern high school curriculums [p. 3].

A possible criticism of combining some, or all, of the different applications into a more effective utilization of the computer is that for each individual application, little theory of implementation has been developed and not an overly impressive amount of evidence has been presented to justify the claims of improved educational effectiveness. The lack of controlled experiments to test computer effectiveness is due to limited availability of computers in secondary schools for experiments and the political or ethical problems of denying a control group the use of the computer. When experiments have been conducted, their generalizability is questionable because of the glamour of the computer causing the "Hawthorne effect" and the high dependency of the success of the experimental group on the computer knowledge and programming expertise of the instructor. However, realizing that constraints on research in computer utilization do exist,
the urgent need from computer educators for effective methodology warrants that such research efforts be continued, improved, and expanded, including research in the effects of combining two or more applications.

Moinar (1971) suggests that "What is needed is not another demonstration of computer-based instruction but a critical mass that is capable of providing a complete system and a total curriculum [p. 64]." He suggests that no less than the cooperation of business, universities, large school systems, and the federal government is needed to develop such an all-inclusive approach. Realizing the magnitude of the overall problem, any report by a single individual could not hope to offer a solution. However, research by a single experimenter can offer some innovative ideas and report some experiences and results that could possibly influence the decisions of future curriculum developers. This will be the purpose and substance of the subsequent chapters of this report.

If a major curriculum area were to be chosen for integrating the various educational capabilities of the computer, mathematics seems most appropriate because of its unlimited opportunities to use the computer in any of the previously described modes; and, within mathematics, few courses are more suited to combine these applications than analytic geometry. In support of this choice, Oettinger (1969) states:

the computer can serve widely shared goals - teaching of analytic geometry - or particular ones - exploring a particular problem. The student may use it to solve prescribed exercises or avail himself freely of as much of its mathematical and graphical power as he is able to use. These are the
possibilities I see [p. 213].

2.10 Summary

A list of some of the issues resolved to date regarding computers in mathematics education might include:

1) Terminal presented drill-and-practice exists and is effective in most cases.

2) Complete courses can be presented via a computer terminal, but TPI is still too expensive for secondary schools.

3) Automated testing is both feasible and effective.

4) Almost any type of information can be presented via a computer terminal.

5) Terminals in secondary schools are too valuable a commodity to be dominated by a few students.

6) Instructional objectives are necessary for generating TPI and organizing instructional management systems.

7) Computers can maintain student records and prescribe remedial and advancement activities.

8) Students at almost all levels can write programs; BASIC is the most accepted language for doing so at the pre-college level.

9) Algorithms are a welcome addition to mathematics courses. The generation of these algorithms allows creative students to create.

10) Most students and teachers like to work with computers but some people can be completely "turned off" by them.

A synopsis of past efforts might be, "Many things can be done with a computer in the classroom."

A statement that might describe the problems yet to be solved is, "How effective are these things that have been done and are they worth it?" This statement is not intended to ignore the research that has been reported to date, but rather to emphasize the
inconclusiveness of many of the results. The measuring of the effectiveness of computer treatments is still hampered by limited sample sizes, limited computer resources, and undeveloped methodologies. Besides determining the effectiveness of computer applications, a few other unresolved issues seem to be:

1) Is there a loss of personality or any other psychological manifestations due to prolonged contact with a computer terminal?

2) Who is going to write and maintain the software needed to support instructional management systems, mathematics information systems, computational subroutine libraries, and demonstrations?

3) Does a student learn mathematics when he writes computer programs? If so, what kind?

4) Assuming programs are to be written, how many should a student be expected to write during a mathematics course? Is it more effective to solve a few complicated problems or many easy problems?

5) Should schools offer separate courses with the computer as the subject of instruction? If so, who should teach these courses and what aspects of computers should be taught?

6) What is the best way for a school to get started using computers?

7) Are integrated systems feasible or are they too demanding of computer resources and teacher time?

In conclusion, the following remark by Thomas Kurtz, the other co-author of BASIC, seems appropriate: "We might say of computing that 'There is much more than meets the CAI' [in Nevison, 1970, p. 3]."

This concludes the review of related literature and research. The following chapter will briefly describe the design of the study.
CHAPTER THREE
DESIGN OF THE STUDY

3.1 Introduction

This chapter will include a discussion of the environment of the study, the computer facilities, the students, and the organization of the analytic geometry course to be taught. Also included will be a description of the specific computer activities and an explanation of how the students were to interact with these activities.

3.2 Environment

Cincinnati Country Day School is a private all-boys school located 20 miles from downtown Cincinnati, Ohio. It is primarily a college preparatory school with excellent educational facilities. Class sizes are small and the credentials of the faculty are impressive.

The students normally attend school from 9:00 A.M. to 4:30 P.M. with the last hour devoted to athletics. The geometry class under consideration was taught the first period of each day, from 9:00 to 9:45 A.M. Most of the 18 students in the class had at least one free period per day during which they could use the computer. In addition, the computer was available for use before and after school and during the lunch period.
The computer room was a few doors down the hall from the classroom and was about the size of a normal faculty office. It had a window air conditioner, two sliding door cabinets for program storage, a work table, two student desks, and two folding chairs in addition to the computer and its teletypewriter (TTY). The computer was situated on a table which also contained a card reader and a paper tape rewinder. Because of the small size of this room, its people capacity was best described by the expression, "Three's a crowd!"

**Computer.** The computer was a Hewlett-Packard 2114C which is considered a "minicomputer." It is a 16-bit word computer with 8,192 words of memory. In minimal configuration its input/output (I/O) capability consists of an ASR-33 teletypewriter manufactured by Teletype Corporation. This I/O device can input or print typed text at 10 characters per second (cps) and can also read or punch paper tape at the same rate. However, since 10 cps is too slow for loading large programs such as the BASIC interpreter, the school added a HP 2748 high-speed paper tape reader which can read at 500 cps. In addition, an HP 2761A Optical Mark Reader was included to increase throughput by allowing high speed card input to supplement the rather slow keyboard input from the teletypewriter. This reader has an input rate of 200 cards per minute and can read data that is either keypunched or marked with an ordinary lead pencil. The software for the computer included a three-pass assembler, a three-pass Fortran compiler, a two-pass ALGOL compiler,
and a BASIC interpreter.\(^1\) The entire configuration rented for $590 per month, including maintenance, on a one-year lease.

**Minicomputer versus time sharing.** The use of a minicomputer is similar to time sharing in that they both usually have typewriter style input terminals; however they differ in the availability of high-speed input-output (I/O) devices such as card readers and high-speed paper tape readers. These I/O devices are not normally used in a time-sharing situation due to the costs of additional communication facilities required for the high information transfer rates. Minicomputers are also usually more reliable than time sharing, have a constant response time\(^2\), and allow unlimited usage.\(^3\) The major disadvantage of the minicomputer is its limited memory capacity. This limitation restricts the size of programs that can be run on the computer and inhibits the use of

---

\(^1\)A **compiler** is a special program that converts a user's program in a problem-oriented language to a sequence of internal instructions which are then executed after the compiler program is purged from the computer's memory. An **interpreter**, also a program, differs from a compiler in that it converts the user's program statements as they are encountered in the execution of the program; consequently it must always be resident in the computer's memory. Interpreters are usually used for one-time student programs whereas compilers are used for production programs that will be used many times. An **assembler** is a program that converts a user's symbolic (assembly language) program into its internal binary form. Assembly language programming is a very tedious and time-consuming process and therefore is only used when there are execution time or memory constraints.

\(^2\)Response time is the time it takes for the computer to respond to a student action at the terminal. In a time-sharing environment this time usually increases as the number of users of the system increases.

\(^3\)Although time-sharing costs can be arranged to allow unlimited usage, usually these costs are a function of computer time used, data transferred, and use of memory capacity.
subroutine libraries and large data banks.\textsuperscript{1}

3.3 Students

The students participated in a two week introductory course in computer problem solving during the early part of the school year. The language used was BASIC and the course materials were prepared by the mathematics and science faculty of the school (Laird, 1970). Although most of the students attempted the programming exercises required of this introductory course, no assessment was made of their programming competence in BASIC; however, some of the students did use the computer during the school year for mathematics and science problems.

The intellectual capacity of this group would be considered well above average if one considers IQ as a valid measure. The mean IQ of the class was 134 with a standard deviation of 9 and a range of 118 to 150. In addition, the completion of a geometry course one month early after having devoted two weeks to learning BASIC is a situation that is not common in secondary schools except with the most able classes.

Because this study was a developmental effort, it was agreed by Mr. Klitz and the author that the students should not be graded on their efforts. This condition was to be emphasized to the students.

\textsuperscript{1}Of course additional memory can be purchased for minicomputers just as high-speed I/O devices can be purchased for time-sharing systems. However, because of the vast combinations of alternative systems, including time-shared minicomputers, each school must perform its own cost-benefit analysis consistent with its instructional requirements and available funds.
during the orientation lecture and reinforced by giving each student an identification number. The key to associating student names with identification numbers was to be kept secure and inaccessible for the duration of the study.

3.4 Organization of the Course

In keeping with the goals of the developmental effort, i.e., to modify an established classroom course, Mr. Klitz and the author agreed that a textbook should be used to help organize the course and to provide student reading material. The textbook chosen was "Contemporary Analytic Geometry" by Thomas L. Wade and Howard E. Taylor (1969). As an aid to course organization the preface to this book contains suggested lesson assignments for courses of 22, 33, and 44 classroom hours corresponding to 2, 3, or 4 college quarter hour credits. Since only 20 class sessions were available, the outline developed by Mr. Klitz and the author (see Table 1) was a slight modification of Wade and Taylor's recommended outline for a course of 22 hours. The most significant modification was replacing the theorem-oriented vector geometry sections of Chapter 7 with the more numerical-oriented solid analytic geometry sections of Chapter 8.

In addition to the four weeks of instruction, two class sessions of the week preceding the course were to be used for a pretest and an orientation lecture, and one class session of the week following the course was to be used for the student evaluation of the course.
3.5 **Computer Activities**

The computerized portion of the course was separated into five principal activities: quiz grading, Terminal Presented Instruction, homework programs in BASIC, record keeping, and report generation. The following paragraphs will describe the required student and

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.1</td>
<td>1.5</td>
<td>1.7</td>
<td>2.1</td>
<td>2.7</td>
</tr>
<tr>
<td></td>
<td>1.2</td>
<td></td>
<td>1.8</td>
<td>2.2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.3</td>
<td>1.6</td>
<td></td>
<td>2.3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.4</td>
<td></td>
<td></td>
<td>2.4</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2.9</td>
<td>3.2</td>
<td>3.4</td>
<td>3.6</td>
<td>4.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.1</td>
<td></td>
<td>3.7</td>
<td>4.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3.8</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>4.4</td>
<td>4.5</td>
<td>4.6</td>
<td>5.1</td>
<td>5.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>6.3</td>
<td>6.4</td>
<td>6.5</td>
<td>8.1</td>
<td>FINAL</td>
</tr>
<tr>
<td></td>
<td>6.2</td>
<td>6.6</td>
<td>8.2</td>
<td>8.3</td>
<td>EXAM</td>
</tr>
<tr>
<td></td>
<td>6.3</td>
<td>6.8</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1

Course Outline: Organized by sections of the text
teacher interface with the computer for each activity.

Quiz Grading. A seven-item, multiple-choice quiz was to be given each session except the first and last; this resulted in 18 quizzes overall (see Appendix A). To record their answers (and the time spent on homework) in computer compatible form, each student was to be given a deck of 18 cards, one for each quiz. These cards (see Figure 2) were prepunched with the student's identification number, quiz number, and BASIC syntax requirements to reduce marking errors. Because of the prepunching, the student only had the responsibility of marking the correct columns of the card with his quiz answers and homework time. However, if a student forgot his prepunched cards, he would have to properly mark all the required columns of a blank card.

![Fig. 2 Sample Quiz Card](image-url)
After class, Mr. Klitz would take the quiz cards to the computer room where he would place a special prepunched card containing the quiz key on top of the student quiz deck and a prepunched trailer card on the bottom. Appendix G contains a diagram of the required deck arrangement. He would then place the quiz deck in the input hopper of the card reader and load the paper tape containing the quiz correcting program into the computer via the high-speed paper tape reader. The quiz program would compare each student's result to the quiz key and enter his score into an internal memory location. The program would also store in the computer's memory the number of students missing each question. This data would eventually be written out by one of the report generation programs.

Terminal Presented Instruction. Ten TPI programs were to be made available to the students. These programs were coordinated with the sections of the text and were designed to furnish 1) a quick review of an old topic for those students having difficulty or 2) a brief introduction to a new topic for those students working ahead. It was not anticipated that a student would need to or want to interact with every TPI program.

The TPI programs were to consist of a series of questions to be answered by the student (see Figure 3). If a student answered a question incorrectly, he would be given the correct answer with a short explanation of how it was obtained. He would then be asked the same question, but with different data, until he either got the question correct or had three tries; he would then be asked the next question. Occasionally, depending upon the interrelationships of the
Q6. IF X IS AN INTGR, THE # OF SOLNS OF X+2<= 7 IS
5
SUM OF ABS VAL OF SOLNS IS
WRONG! 5 SOLNS SUM ABS VAL= 6

Q7. GRAPH OF CLUSO INTRVAL <-4;3> IS
-----------------------------
-5 -4 -3 -2 -1 0 1 2 3 4 5
GRAPH OF OPEN INTRVAL (-1;5) IS
-------------------------------
-5 -4 -3 -2 -1 0 1 2 3 4 5
THE GRAPH
-------------------------------
-5 -4 -3 -2 -1 0 1 2 3 4 5
IS REPRESENTED BY:
1. <-2 ; 4 >
2. (-2 ; 4 )
3. <-2 ; 4 >
4. ( 4 ; -2 )
WRONG! ANS=3

THE GRAPH
-------------------------------
-5 -4 -3 -2 -1 0 1 2 3 4 5
IS REPRESENTED BY:
1. <-5 ; 0 >
2. <-5 ; 0 >
3. <-5 ; 0 >
4. ( 0 ; -5 )
CORRECT! NO. 1

10 WRONG FOR TRY 1 OF CAL # 1
READY

Fig. 3 Questions from TPI No. 1
questions, the program would recycle two or three questions back.

Each program was designed to have at most 10 questions and to require no more than 5 minutes terminal time. Upon completion of the program, the total number of incorrect answers would be printed at the TTY and entered into the student's records in the computer's memory. If desired, a student could retake the same TPI program up to a limit of three times.

Homework programs in BASIC. Student homework for the course would consist of two problems to be programmed each night, resulting in 36 programs overall. These problems (see Figure 4) were related to the section of the text that were to be covered during the lecture preceding the assignment. It would be suggested that students who were having trouble do only one problem per night whereas the "programming experts" would be encouraged to work as far ahead as possible. The homework programs were to be written in BASIC and were chosen so that their solution required no more than 20 BASIC statements. It was anticipated that the students would program the problems at home on the mark-sense cards. Hopefully, this would reduce the congestion at the TTY terminal and allow for greater throughput.

The student program was to become the second part of a three-part program. The first portion of this program would generate data to be manipulated by the student's program. The results of the student's program would then be checked by the third portion of the program. This arrangement is diagrammed in Figure 5. Just as each homework problem requires a different algorithmic solution, each problem would also require its own unique data generation and result checking.
Computerized Analytic Geometry

Programs due 20 Apr 71

1. INPUT: II = number of elements in set A
   N2 = number of elements in set B
   A = array of elements in set A
   B = array of elements in set B

   OUTPUT: E = 1 if A=B
            = 0 if A ≠ B
   N3 = number of elements in set A ∩ B
   C = array of elements in set A ∩ B
   N4 = number of elements in set A ∪ B
   D = array of elements in set A ∪ B

   STATEMENT NUMBERS: £1000 and <1200

   NOTE: For this program and all subsequent programs assume all arrays have been properly dimensioned and that an ENE statement is unnecessary. Terminate all programs with a GO TO 9000. All variables ending in a 9 (ie. F9) and the U and V arrays are not to be used by the student.

2. INPUT: X1 = coordinate of point P in a one-dimensional coordinate system
   X2 = coordinate of point P
   X3 = coordinate of point P

   OUTPUT: D = the minimum of all possible directed distances
            E = the maximum of all possible undirected distances

   STATEMENT NUMBERS: £1200 and <1400

Programs due 21 Apr 71

3. INPUT: II = number of elements in set A
   A = array of elements in set A

   OUTPUT: N2 = number of elements in A ∩ A
   B = array of left members of ordered pairs of A ∩ A
   C = array of right members of ordered pairs of A ∩ A

   STATEMENT NUMBERS: £1400 and <1600

4. INPUT: II = number of elements in set A
   A = array of elements in set A
   FIA(X) = arbitrary predefined function

   OUTPUT: N2 = number of elements of R = (x,y) | x ∈ A and y = FIA(x) where R is a relation in A X A
   B = array of left members of ordered pairs of R
   C = array of right members of ordered pairs of R

   STATEMENT NUMBERS: £1600 and <1800

Figure 4 Sample homework problems.
portions. These portions, called "check programs," were to be generated by the author and stored on paper tape with a different tape for each problem. Appendix B contains a listing of the homework assignments, a sample check program, and some sample correct solutions.

The remaining paragraphs of the section will discuss the desired procedure for running the homework programs. This procedure is described in flowchart form in Figure 6. The initial step would be to have a student operator load the high speed paper tape reader with the paper tape for the given problem and then load the card reader with the student's program deck. If the student's deck had BASIC language
Fig. 6. Procedure for running homework programs
errors, they would be listed on the TTY for his immediate correction, unless, of course, there were too many or the student couldn't understand the error message. In these situations the student would sign off the TTY and correct his cards while someone else was running their program.

Assuming all BASIC languages errors were corrected, the student would type RUN which would result in his being asked to type in his student identification number. This number would be used to check his programming record to determine if he had already tried the given program seven times. If so, a message would be printed telling him that he had already had seven tries and that his program would not be checked. If the number of tries was less than seven, the check program would immediately start generating sample data.

For some problems, the check data would be randomly generated; while for others, particularly problems with many conditional situations, the check data would be chosen to test all possible conditions. If the student's program had an algorithmic error for any set of check data, the "result" section of the check program would list the check data, the student's results, and the correct results (see Figure 7). The check program would then increase by one the number of tries for that student for the given problem in the programming section of the student's record. Unless the errors were trivial, the student would be expected to give up the terminal when this type of error occurred.

If the student's program was correct for the first set of test data, the result checking portion would print out "OK1" and then cycle back through the second set of test data. Most programs would have
between five and seven sets of test data. If the student got correct answers for all sets of test data, the program would print out "PROG CORRECT" and update the students programming record to indicate that he had correctly solved the problem.

Fig. 7 Sample output from check program No. 2

Record keeping. A section of the central memory of the computer was to be dedicated for student records for the duration of the study. This storage area contained student quiz, programming, and TPI records and was to be non-accessible to the ordinary BASIC programmer. It could only be entered by calls from the quiz correction, TPI, homework checking, and report generation programs.
Because of memory limitations, quiz item analyzes, student homework times, and sessions numbers that students used the homework checking and TPI programs were not kept in computer memory. This data could be obtained, if necessary, from the student quiz cards and the daily teacher summary printouts. Data concerning the amount of computer contact time for each student was to be obtained from times recorded on daily sign-in sheets. A copy of one of these sheets can be found in Appendix G.

Reports. Two standard reports were to be generated: a teacher’s summary (see Figure 8) and an individual student report. The teacher's summary was to be generated after the daily quiz cards were loaded into the card reader. Because of the size of the program, it was separated into two parts. The first part checked the quiz cards for marking errors and, after all errors were corrected, printed the number of student's missing each of the seven items on the quiz. This information was followed by the student's cumulative quiz record which listed each student's score on each quiz. The last item printed was the average time spent by the class on homework.

The second part of the teacher's summary report, which required loading a second tape, listed the student's cumulative programming records, TPI records, and suggested activities. The programming record indicated the number of times each student tried to solve each problem and whether he had obtained a correct solution. The TPI records indicated the number of times each student tried each TPI program and the number of incorrect answers on the latest try. The prescriptive portion of this report listed for each student whether he should take
### RESULTS FOR 19 STUDENTS TAKING QUIZ NO. 6

<table>
<thead>
<tr>
<th>QUESTION NO.</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>NUM. WRONG</td>
<td>3</td>
<td>4</td>
<td>6</td>
<td>13</td>
<td>11</td>
<td>6</td>
<td></td>
</tr>
</tbody>
</table>

### STUDENT CUM QUIZ RECORD

<table>
<thead>
<tr>
<th>STUDENT NO.</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>NUM. QUIZ</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
</tbody>
</table>

### AVG TIME SPENT ON HW 43.8778 MIN

### STUDENT CUM PROG RECORD

<table>
<thead>
<tr>
<th>STUDENT NO.</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>NUM. TRIES</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

### STUDENT CUM CAI RECORD

<table>
<thead>
<tr>
<th>STUDENT NO.</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAI NO.</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
</tbody>
</table>

**Fig. 8 Teacher daily summary report**
a particular TPI or try to program a particular problem. This pre-
scription depended upon his latest quiz score, programming record and
TPI record. The logic of this prescription process is described in
flowchart form in Figure 9.

The student's report required a special program to be run after
the student had one of his homework programs checked. This report
presented the same information as the teacher summary except that it
was more current and was designed for the individual student. The
student would have some control over the amount of information
printed. His first choice upon entering the program would be to in-
dicate whether he wanted his quiz score for that day. He would then
be offered the choice of having his record-to-date printed; this
record included his quiz scores, programming tries, and TPI record.
Upon completion of this program, the student was given a message pre-
scribing his next activity. This program used a somewhat different
prescription algorithm than the teacher summary because it was
anticipated that the student would interact with this program more
than once a day and therefore needed a prescription that accounted for
the student's activities since taking the morning quiz. The logic of
this algorithm is imbedded in the flowchart of the student record
program in Appendix D.

3.6 Daily Operating Procedures

This section will discuss both classroom and computer room opera-
ting procedures.

Classroom. Each class session of the course, except the first, was to
begin with a seven item quiz. A maximum of 10 minutes was to be
Fig. 9 Logic of prescription process

1. If QUIR score > 5?
   - NO
   - YES
     - TPI FOR THIS SESSION BEEN TAKEN?
       - NO
       - YES
         - < 5 WROUGHT ON LATEST TRY?
           - NO
           - YES
             - TAKET TPI FOR THIS SESSION

2. LET "L" = NUMBER OF PROGRAM REQUIRED FOR CURRENT SESSION
   - NO
   - YES
     - HAS ANY PROGRAM WITH NUMBER > L BEEN COMPLETED?
       - NO
       - YES
         - LET "J" = NUMBER OF TPI ASSOCIATED WITH NEXT CLASS SESSION

3. HAS TPI # J BEEN TAKEN?
   - NO
   - YES
     - < 5 WROUGHT ON LATEST TRY?
       - NO
       - YES
         - TAKET TPI # J 3 TIMES?
           - NO
           - YES
             - INCREASE J BY 1
   - YES
     - TAKET TPI # J

4. J < 10
   - YES
   - NO
     - DO HOMEWORK PROGRAM # L

NOTE: THIS ILLOGICAL PATH WAS NOT NOTICED UNTIL THE STUDY WAS COMPLETED.
allowed for the completion of the quiz with the remainder of the session devoted to a review of the quiz and instruction in the designated sections of the text. It was anticipated that some classroom time would be needed for explaining algorithms but no lecture time was to be allocated for debugging individual programming problems.

**Computer room.** Instruction sheets on the individual computer activities were to be supplied to the students during the orientation lecture. These instructions assumed the availability of student operators in the computer room to run the quiz correcting programs for Mr. Klitz and to assist the students in loading their programs. An additional function of the operator was to encourage the students to record their activity on daily sign-in sheets posted in the computer room. Copies of the sign-in sheet and the student and operator instructions are available in Appendix G.

The students were to use the computer on a first-come, first-served basis. A schedule was not considered necessary or desirable because it was anticipated that the students would only be at the terminal for the short TFI programs or for the few minutes it took to load and check their homework programs.

### 3.7 Tests and Evaluation

The course was to be evaluated by assessing student achievement; determining the amount of student time expended; and obtaining student, teacher, and experimenter observations. Student achievement was to be assessed by the daily quizzes, TFI scores, homework results, and a final exam.

Since the daily quizzes were designed to assess the learning of
the course material as it was presented, it was decided that, if possible, the final exam should be a standardized test or, at least, a test that had been validated by comparing it to a standardized test. The test chosen was selected from the instructor's manual accompanying the programmed text Analytic Geometry by Thomas A. Davis (1967).

A pretest on analytic geometry and BASIC was also to be given during a class session of the week preceding the course. Although no experimental comparisons were planned, the use of a pretest was felt necessary to help in the process of formulating hypotheses for future experiments.

Students observations were to be obtained from comments on the daily sign-in sheets and by having the students complete a questionnaire after the course terminated. A copy of this questionnaire is available in Appendix F. In essence, the questionnaire assessed the students' attitudes towards the methodology used and asked their recommendations for future improvements. The teacher's observations were to consist of his personal thoughts as the course progressed and any significant anecdotal comments that the students might make. The experimenter's observations were to consist of a record of significant milestones and an annotation of any developmental or implementation problems encountered.

In summary, this chapter has presented the planned procedures for implementing the study. The following chapter will describe in more detail how the planned activities were actually implemented and relate the significant problems encountered during their implementation.
4.1 Introduction

This chapter will explain some of the procedures used in implementing the course described in the previous chapter and describe some of the unanticipated problems that occurred during the implementation. Each of the computer activities will be discussed in a separate section. The first portion of these sections will include an overview of the effort required and problems encountered; the second part will include more technical details of the computer functions. To set the significant events in proper time perspective, a chronology is presented in Table 2.

4.2 Orientation Lecture

An orientation lecture was given by the author during one 45 minute class session on the Friday preceding the planned starting day. The lecture was organized to include instructions for the activities of the course and an introduction to computer concepts; unfortunately, there was only time to discuss the course instructions. Before the specific activities of the course were explained, the students were guaranteed that their current geometry course grade would not be affected. They were also told that the course would be difficult, but that this difficulty could be considered a challenge for them to prove to themselves what their capabilities were. However, they were cautioned not to overextend themselves to the degree that they would be slighting their other courses. In addition, the students were
<table>
<thead>
<tr>
<th>Date</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>September 1970</td>
<td>Geometry class participates in the DAV program: An indoctrination course in BASIC.</td>
</tr>
<tr>
<td>Wednesday, 17 Mar 1971</td>
<td>Meeting with Ralph Kiltz at Cincinnati Geometry Day School (CUG) to establish course content, meet textbook and supporting course outlines.</td>
</tr>
<tr>
<td>Saturday, 27 Mar 1971</td>
<td>First hands-on experience with CUG equipment. Tried a trial TPI program. Discussed record keeping strategy with David Talbot, a computer teacher and the junior's own computer user, and with Jeff Spain, a member of the junior class and the school's computer expert.</td>
</tr>
<tr>
<td>Sunday, 4 Apr 1971</td>
<td>Letter sent to Jeff Spain defining the required assembly language record-keeping subroutines.</td>
</tr>
<tr>
<td>Monday, 5 Apr 1971</td>
<td>Administration of protest.</td>
</tr>
<tr>
<td>Friday, 16 Apr 1971</td>
<td>Orientation lecture.</td>
</tr>
<tr>
<td>Saturday, 17 Apr 1971</td>
<td>Call from Mr. Kiltz reporting that Jeff could not get the record keeping routines debugged. Decision to postpone the course for a week.</td>
</tr>
<tr>
<td>Monday, 19 Apr 1971</td>
<td>Planned starting day.</td>
</tr>
<tr>
<td>Thursday, 22 Apr 1971</td>
<td>Assembly language programs debugged.</td>
</tr>
<tr>
<td>Monday, 26 Apr 1971</td>
<td>Actual starting day.</td>
</tr>
<tr>
<td>Friday, 30 Apr 1971</td>
<td>No class, school testing.</td>
</tr>
<tr>
<td>Monday, 10 May 1971</td>
<td>Commitment to student anonymity was broken to identify the non-programmers and encourage them to at least try the TPI programs.</td>
</tr>
<tr>
<td>Thursday, 13 May 1971</td>
<td>Mr. Kiltz began to use the overhead projector to increase the amount of material presented in classroom lectures.</td>
</tr>
<tr>
<td>Friday, 21 May 1971</td>
<td>Last day of course.</td>
</tr>
<tr>
<td>Monday, 24 May 1971</td>
<td>Final exam.</td>
</tr>
<tr>
<td>Tuesday, 25 May 1971</td>
<td>Student evaluation of course</td>
</tr>
<tr>
<td>Friday, 26 May 1971</td>
<td>Lecture on computer concepts given by the author.</td>
</tr>
</tbody>
</table>
encouraged to voice their comments or criticisms at any time to Mr. Klitz or the author.

After the introductory remarks were completed, each student was given a deck of quiz cards, a quiz instruction sheet, a programming instruction sheet, a TPI instruction sheet, and two sheets containing the first eight homework problems.1

The quiz-taking procedures were discussed first. Each student's quiz deck contained a blank cover card and a card for each of the eighteen quizzes. The student's identification number and the quiz number were prepunched and typed at the top of the card. After the quiz instructions were read by the author, each student was asked to mark the blank card with sample quiz scores of 3, 2, 5, 0, 1, 4, 3 and a sample homework time of 090 minutes. Mr. Klitz and the author checked each student's card to guarantee that the proper columns had been marked. This activity took considerably more time than expected (approximately 15 minutes). After all students had marked their cards correctly, each was asked to sign his name on the back of the card. To emphasize the commitment to student anonymity, the author would not let Mr. Klitz collect these cards but rather collected them himself and, in the students' view, put them in his inside jacket pocket.

The programming instructions were read next and the reasons for the required programming conventions were explained. The students were encouraged to flow chart their programs first and then write them on scratch paper before marking the cards. They were also encouraged to

---

1The remaining homework sheets were given to the students the first day of the course.
generate some sample data and use it to evaluate their algorithm before trying to run the associated program on the computer. To facilitate debugging, the advantages of using "PRINT" statements in their programs for listing intermediate results was also discussed. The limit of seven tries for each program was emphasized and the students were told that if they did not get a correct solution after seven tries, they could ask Mr. Klitz to let them look at the author's version of a correct program for the problem.

The remainder of the lecture briefly discussed the use of the TPI programs, the interpretation of the student reports, and the sign-in procedure for using the computer.

The lecture ended with the author asking the students to be patient with operational malfunctions and to report any incorrect student data, or any errors in homework checking or TPI programs, to Mr. Klitz.

4.3 Quiz Correction and Report Generation

Following the orientation lecture, Mr. Klitz was given nineteen copies of each quiz, one for each student and one for himself. He was also given a prepunched deck of cards containing the eighteen quiz keys and the required control cards, and a diagram for proper quiz deck set-up (see Appendix G).

Because student operators didn't materialize, Mr. Klitz had to load the quiz decks and the teacher report programs into the computer. For the first few days of the course this was a frustrating experience because of his unfamiliarity with the report programs and because a few students would forget their prepunched cards and have to fill them out completely by hand, always incorrectly. In fact, it was not until
the fourteenth session that there were no errors in the student quiz cards. Mr. Klitz commented on the seventh day that, "I find I'm spending approximately 1 period/day average just running quiz programs; all due to student errors in marking cards." Because of this inordinate time demand, the running of the quiz program was occasionally delayed, and at one time near the end of the course, was two days behind.

Occasionally the prepunched quiz keys supplied in advance to Mr. Klitz would have an incorrect answer for one of the questions. This error was easily remedied by pencil-marking a new quiz key with the correct answer. The teacher reports became rather lengthy toward the end of the course and required up to 15 minutes for printing. Originally this report was to be kept in Mr. Klitz's possession but after it was determined that the students weren't using the student report program, it was decided to post the teacher report on the classroom bulletin board. Unfortunately, this compromised the student anonymity because it didn't take the students very long to break the code.

Technical details. All quiz questions, except the first, were adapted from problems at the end of the textbook sections discussed in the previous class session. The first quiz question usually was a review of material covered two sessions prior to the quiz. Most of the questions were five-alternative multiple-choice with an occasional true-false item included. A conscientious attempt was made to supply alternative choices that were the results of applying reasonable but erroneous concepts.
Naively, the author had the quiz number and class date typed on each quiz. This caused some confusion when the course was delayed and one class session was missed. It would have been better to have left the date off or used the statement "QUIZ FOR SESSION ---," a technique the author has adopted in his Computer Capability and Utilization Course.

There were three programs involved in quiz correcting and reporting activity; two for the teacher report and one for the student report. The first program for the teacher report consisted of 84 BASIC statements and performed the tasks of 1) checking each student's quiz answer against a quiz key, 2) updating the student's quiz record, 3) printing the number of students missing each quiz question, 4) calculating and printing the average homework time, and 5) printing each student's cumulative quiz record. The input to this program was the deck of mark-sense cards containing the quiz key and the student's quiz cards. Before processing any of the quiz data, this program made editing checks verifying that 1) the first card of the deck was a quiz key, 2) each student's identification (ID) number was within the range 1 to 18, and 3) the quiz number on each card agreed with the quiz number on the quiz key card.

To enter the quiz data into this program, the BASIC statements READ and DATA were used. By having each student's identification number + 9000 prepunched in the BASIC statement number field, it was possible to guarantee unique DATA statements. This allowed the quiz cards to be entered in any order and eliminated any requirement for a fixed number of cards to be entered each session. A flow diagram of this
program and the other two reporting programs is available in Appendix D.

A second program was necessary for the teacher report because the combined program would have been too large for central memory. The separation point between the two programs was chosen to minimize the amount of data to be kept common to both programs. This resulted in only one data element being kept in common; the quiz number. The COM statement of BASIC was used to effect this communication. This second program consisted of 115 BASIC statements and performed the tasks of printing each student's cumulative homework and TPI records and his prescribed activity for the next session. The program accessed the student data base for student information and contained a table of TPI numbers associated with each session to assist in the prescriptive process. Such a table was not necessary for homework problem numbers because they were linearly related to the session number.

Although the above two programs were designed to print only significant data, the resulting teacher report for the last session consisted of 218 lines of output. At an average of 60 characters per line and a typing rate of 10 characters per sec (the normal teletype rate), this report took 21.8 minutes to print. One alternative for reducing the length of this report would have been to generate only daily results, but this would have required the ability to determine which homework programs or TPI were run during a given day. This data was not available, and, even if it were, it is doubtful whether the time saved by using daily reports would have been worth the loss of convenience and information.
The third program, the student report program, consisted of 83 BASIC statements. This program could be run following any homework checking program and had the student’s identification number and program number transferred in common from the check program. Unfortunately, this program was not used by any of the students.

The three reporting programs were punched on paper tape and labeled Q1, Q2, and SR. Mr. Klitz had back-up copies of all three tapes.

4.4 Homework Checking

One of the considerations in developing a procedure for checking student programs was to determine the form of interface. Alternatives such as checking a generated output tape or having the students call a checking program were discarded in favor of the approach of embedding a student’s program within the check program. The student’s program would use test data stored by the check program in memory locations associated with the input variables and would place the results to be checked in memory locations associated with the output variables. In addition to its checking advantages, this procedure facilitated the presentation of the problems themselves (see Figure 4).

The homework problem descriptions required the students to use specified variable names and to keep their program statements within a given range. The specified variable names for input and output were chosen to be as consistent as possible to the variable names used in the associated sections of the text. In addition, some specific variable names were not allowed because of the possibility of

---

1 This procedure was inadvertently violated in the description of problems 19 thru 23.
overwriting the check data. The constraint on program statement numbers reduced the probability of a student overwriting statements in the check programs and, since each program had a unique range of permissible statement numbers, allowed the students to use previous programs as subprograms for later problems. A copy of all homework problem descriptions is available in Appendix B. Unfortunately, not all problem descriptions were completely explicit; those problems requiring input or output of angular measurements didn't specify whether the angles were measured in degrees or radians. This oversight was corrected before any students tried those problems. The problem on matching activities to shifting dates encountered with the quizzes also occurred with the homework assignments.

Probably one of the major errors in the selection of the homework problems was to make the first problem one of the most difficult. (To add to its difficulty, a few of the students who had programmed the problem correctly were required to rewrite their solutions because they used instructions from the matrix packages which had been deleted). The difficulty of this first problem was probably a contributing factor in the reluctance of students with poor programming backgrounds to try any problems. Mr. Klitz commented during the second day of the course that, "[I] have observed that five of our students really know little about programming. Apparently little effort was put forth by these students early this school year."

In all, 33 of the 38 programs were tried by at least one student. One of the most common student complaints was errors in the check programs. Although such errors only existed in two or three programs,
the students were quite irked with them when they occurred, probably because they were being charged with an incorrect program try in their student record.

The obsession with obtaining a good student record (probably because their records were posted on the bulletin board) and/or the desire to have more than seven tries at a program caused some students to alter their programming records by modifying the check programs. One of the techniques used to do this was to replace the call to the record keeping program with a transfer to the end of the check program. The students discovered this "trick" by the fifth class session. Another manifestation of the desire for a good record was the accusation by one student that some of the students copied programs and, in one case, even tried to buy correct solutions from other students. Such a charge would be difficult and undesirable to verify.

Check programs and sample correct solutions were written for all 38 problems. The average time for generating the check program and sample correct solution was 100 minutes. Each program performed five major activities: 1) updating the student homework record, 2) generating test data, 3) generating correct answers, 4) comparing student answers to correct answers, and 5) printing out the results of the comparison. The technical details of these activities and the procedures for actually writing the check programs will be described in the following paragraphs.

Updating student homework records. After a student loaded the check program for his problem via the high speed tape reader and his program on cards via the card reader, he was asked to type in his student ID
number. If the number he typed was less than 1 or greater than 18, he was asked to retype his student number. Once his number was accepted, the program checked his record to ascertain if he had already tried the program seven times. If so, "TRIED 7 TIMES ALREADY" was printed and the program was stopped. If not, "1" was immediately added to the number of tries in his record. If the student didn't get the program correct, his record was not accessed again; however, if he did get the program correct, "0" was added to his number of tries. ¹

Before the second call to his record occurred, the student's ID number and his number of tries were again checked for being in the proper range. This was done to eliminate the clobbering of the data base by a student who inadvertently used one of these variables in his program. Unfortunately this checking was not foolproof. A better technique would have been to store the initial values of the program number, student ID number, and number of tries in additional backup locations and then compare the present and backup values before calling the record programs.

Generating test data. Two methods of generating test data were used; fixed and random. The "fixed" method used the READ statement to load test values stored in a DATA statement. These values were chosen to guarantee that all options of a problem that required categorization were tested. As an example, problem ten requests that the student's program identify if four points form a square, rhombus, rectangle, parallelogram, isosceles trapezoid, kite, or none of these. The data

¹Any number greater than 7 for a problem in the student homework record indicated that the student correctly programmed that problem.
stored in the seven DATA statements for this problem consisted of the coordinates of four points that formed each of these figures.

The "random" method used the random number generation capability of BASIC. The ranges of the random numbers were chosen to incorporate all possible conditions of input data for those problems that required numerical solutions rather than categorization. An example of this method is the check program for problem nine. Problem nine requires the student's program to generate the length of the three medians of a triangle, given the coordinates of the vertices as input. Table 3 contains the statements used to generate the input coordinates.

<table>
<thead>
<tr>
<th>BASIC statements</th>
<th>range of random integers</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 DEF FNZ(X) = INT(RND(0)*X+1)</td>
<td></td>
</tr>
<tr>
<td>500 LET X1 = FNZ(10)-5</td>
<td>-4&lt;X1&lt;6</td>
</tr>
<tr>
<td>510 LET X2 = FNZ(5)-3</td>
<td>-2&lt;X2&lt;3</td>
</tr>
<tr>
<td>520 LET X3 = FNZ(15)-8</td>
<td>-7&lt;X3&lt;8</td>
</tr>
<tr>
<td>530 LET Y1 = FNZ(-10)+3</td>
<td>-6&lt;Y1&lt;4</td>
</tr>
<tr>
<td>540 LET Y2 = FNZ(20)+10</td>
<td>11&lt;Y2&lt;31</td>
</tr>
<tr>
<td>550 LET Y3 = FNZ(-5)-3</td>
<td>-7&lt;Y3&lt;-2</td>
</tr>
</tbody>
</table>

Statement 20 in the above table defines a function that will generate a random integer in the range 1 to X+1. By calling FNZ from statement 500 with the expression FNZ(10), a random integer in the range 1 to 11 is generated. By subtracting 5 from the value obtained,
XL will be a random integer in the range $-4$ to $6$. The ranges of the other variables are included in the table.

**Generating correct answers.** Because the correct answers for programs with fixed input data were used to help generate the input data, they were known in advance of the running of the program. These answers were conveniently placed in the same DATA statements as the input data and were entered into the program via the same READ statement. The correct answers for the programs with random input data were generated by incorporating a "correct" version of the required student program into the check program. This "correct" solution generated the values that the student's results were compared to.

The variable names for the correct answers were the same as the requested output variable names except that they had "9" appended to them; i.e., if the required student output variable name was B, the correct solution would be stored in B9. The U and V arrays were used for storing the correct solutions for those programs requiring arrays as output. This programming convention constrained the students from using variable names ending in 9 or the U and V arrays.

**Comparing student answers to correct answers.** Some of the student programs required calculated values as output. Since no two programmers normally write identical programs for the same problem, it was possible that correct student answers would differ from the answers generated by the check program. This would occur when different combinations of arithmetic processing resulted in different round-off errors. To guard against the chance of a correct student answer being rejected, student solutions were only rejected if the absolute value of the
difference between the student solution and the check solution was greater than $10^{-4}$. The value of $10^{-4}$ was chosen because the minicomputer used can only guarantee six decimal digits of precision. In retrospect, it probably would have been more consistent to compare the relative difference (absolute difference divided by true value) to $10^{-4}$ rather than the absolute difference.

Some of the homework problems requested arrays of data as output. The checking of these arrays against the correct arrays was not a simple one-to-one comparison because the student's array could be a permutation of the correct array. The procedure used for this checking involved comparing every element of the student's array to the check array and, if no match was found, rejecting the student's solution. In retrospect, this test was necessary but not sufficient. The definition of "equal sets" implies that, additionally, each element of the check array must be compared to the student's array and likewise rejected if no match was found.

The problems that required students to categorize the input data were the easiest to check. No precision considerations or equality of sets was needed. Some problems required multiple outputs; for these problems the logical operators "AND" and "OK" were extremely useful in the checking process.

Printing out results of the comparison. If the student's answers were correct for the first set of test data, the check program printed out "OK 1" and looped back to the statements that generated the test data. If the student's answers were correct for the second set, "OK 2" was printed, etc. until all required sets of test data were used. If the
student's program was correct for all sets, "PROG 1 CORRECT" was printed. Since the programs usually ran very fast, the printing of "OK 1", "OK 2", etc. was rather impressive. However, if a student's program was incorrect for any set of test data, the student wasn't very impressed when "PROG 1 IN ERROR" was printed out. After this error message was printed, the check program printed out the input data and then, in two columns, printed out "YOUR ANSWER" and "CORRECT ANSWER" for each of the output variables (see Figure 7).

Writing check programs. All check programs and correct solution programs were written by the author at his home away from the computer. This was done to insure that they were as correct as possible before running on the computer. From the author's experience in programming, he has learned that minutes spent in initial program writing and hand checking save hours in program debugging. The average time spent writing these programs was 36 minutes; debugging time averaged 64 minutes.

The first few check programs were debugged on a Hewlett-Packard 2115A at Wright-Patterson Air Force Base. These programs were run in the evening when the computer was normally idle. The use of this computer, which was compatible to the HP 2114B, eliminated much travel time from the author's home to OCDS which was approximately an hour and 15 minute drive each way. The only disadvantage of this approach was that the calls to the student record programs couldn't be used because the HP 2115A didn't have the record programs stored in it. These calls were added to the paper tape version of the program after it was debugged by using the "local" mode of teletype.
One technique that saved a considerable amount of typing time was the preparing of a paper tape containing statements that were common to all of the check programs. This tape, called the "start tape," contained 30 BASIC statements and was read into the computer prior to typing any of the statements of the check program. Considering that the check problems ranged in length from 42 to 86 BASIC statements with a mean of 55, the use of the start tape reduced the typing time of a program by more than half. In addition to the writing, typing, debugging, and punching out of the check programs, the same procedure had to be followed for each of the sample correct solution programs. Unfortunately a start program could not be used for the correct solution programs.

The only major problem encountered in debugging the check programs was the author's occasional use of the same variable name at two places in the program. This error played havoc with the student database until it was corrected. In retrospect, a safer convention for selecting variable names would have been to use one appendage (possibly 7) for correct solutions, one (possibly 8) for variables local to the check program, and one (possibly 9) for calls to the student record programs. This would have further restricted the variable names available to the students (BASIC only allows an alphabetic character or an alphabetic followed by a numeric as valid variable names) but they would still have $26 \times 7 = 182$ variable names to choose from, certainly more than should ever be necessary. This does, however, emphasize one often encountered criticism of BASIC: that of a too limited choice of variable names.
Storing the check programs. After each check program was punched out onto paper tape, the tape was rewound and the problem number was placed on the tape leader with a felt-tip marker. The tape was then placed in a 2"x2" cubicle in a cardboard box located on the cabinet in the computer room. Normally paper tape programs are stored in ointment tins or individual cardboard boxes but after calculating a total cost of approximately $18 for the necessary tins to hold all the programs, the author decided to construct two cardboxes with 36 cubicles each. Each cubicle was labeled according to the program that it contained.

Although care was taken in designing a proper storage container for the paper tapes, the author was still concerned about the fact that many students were handling the only copy of the paper tapes. To reduce the possibility of a destroyed program, duplicate copies of the paper tapes were punched and given to Mr. Klitz. However, since the punching of a check program tape took about five minutes, the second punching soon became a nuisance, especially when the program was later found to be in error and had to be corrected and repunched. Therefore, the punching of duplicate check program tapes was discontinued after problem number 16.

4.5 TPI Programming

BASIC was chosen as the language for TPI although Fortran was considered. The programming of the TPI segments was a very tedious task requiring an average of 89 minutes for question development and programming, and an additional 179 minutes for typing, debugging, and punching out a paper tape. Probably the biggest disappointment with
the use of a minicomputer for TPI was its limited memory capacity. This resulted in a reduction in the number of questions that could be asked and a deterioration in the readability of the questions (see Figure 3).

TPI programs were not tried very much at the beginning of the course. This was probably due to an overemphasis on homework problems by some students and a reluctance to consume computer time, or a fear of interacting with the programs, by other students. Approximately half-way through the course, after it was determined that some students were not using the computer at all, Mr. Klitz asked for the names of the non-participants. He talked with them about their lack of computer activity and encouraged them to at least try the TPI. After this time, the TPI programs were used more heavily and by most of the students.

The following paragraphs will discuss the rationale for choosing BASIC over Fortran and will describe some of the technical details of selecting the questions, analyzing the student responses, updating the student TPI records, and writing the programs. This section will conclude with a few observations on the use of HP BASIC for TPI.

Initial choice of language. A desirable characteristic of a TPI author language is the capability of reading in alphabetic characters and performing logical comparisons between the input character strings and character strings internal to the program. Some versions of BASIC have this option. In these versions the character string arrays are stored in locations whose variable names end with a $, i.e., A$. Unfortunately, Hewlett-Packard's version of BASIC did not have this capability. As an alternative the author considered the use of
Fortran which normally allows character input by specifying the data type in a FORMAT statement; however, the H-P version of Fortran didn't have this capability either. Even if Fortran would have had this capability, it is doubtful that it would have been used because of the difficulty of correcting program errors and because Cincinnati Country Day School's version of Fortran had not been checked out.

One advantage of Fortran that didn't become evident until the first TPI was written in BASIC was the potential increase in available storage that might have been realized. This available storage area was the area normally consumed by the BASIC interpreter. After a Fortran program is compiled, the compiler no longer needs to reside in central memory and therefore frees more storage for the user program. If, however, Fortran would be considered for future applications because of this potential memory increase, precaution must be taken during loading and compilation to avoid destroying the student records and record keeping programs which must be permanently resident in the computer.

Selection of questions. The TPI questions were chosen to explain the most important concepts in the sections of the text that were to be covered. Since each presentation was limited to seven or eight questions, the seven or eight most important concepts of the sections were determined before any TPI questions were programmed.

The original strategy of question design was to present a question to the student and, if missed, explain in textual form how the correct answer was obtained. After this explanation the question would be presented again but with different data. To test this
strategy, a few questions from the first TPI program were presented to a person who was unfamiliar with the course material. After observing the bewilderment of this subject when presented questions about which he knew nothing, the author revised the question format so that the textual material was presented before the question. This revision may have been a mistake depending on whether a student was taking the TPI to learn new material or to review old material. Since most students eventually used the TPI for review, the original strategy would probably have been more appropriate. Actually this issue almost became academic because of memory constraints that severely limited the amount of text presented.

Analyzing student responses. The data used in the TPI questions were principally random integers generated in the TPI program. A student's response to these questions was compared to correct answers calculated in the TPI program. If his response was incorrect, the student's number of incorrect responses was incremented and he was presented the same question again but with different random data. If the student missed the same question three times, he was presented the next question in the program. A few of the questions were programmed to transfer the student two questions back if he answered incorrectly.

Since alphabetic input was not available, questions requiring names or symbols as input were usually presented in multiple-choice format. To eliminate programming statements, the multiple-choice response was kept constant for each presentation of these questions even though the data presented to the student varied; it would have been interesting to find out if the students caught on. For those questions
that required sets of numbers as answers, the sum of the numbers in the set was requested. This eliminated the problem of a variable number of input items and also the problem of comparing permutations. Some questions required more than one response; these questions were only considered answered correctly if all responses were correct.

Updating student TPI records. When a student loaded a TPI program, the first message printed was the TPI number followed by a request for the student's ID number. This number was checked for the proper range (0-18) and then was used to retrieve the student's TPI record for that particular TPI program. If the student had already tried the program three times, "4th TRY" was printed out and the program stopped; otherwise he was presented the first question. After the student completed the last question of the TPI program, his number of tries was incremented and entered, along with the number of incorrect responses, into his record. Special appendages, such as those used in the check program, i.e. ending in 9, were not used for TPI program variables that accessed student records. This was poor planning and resulted in the occasional clobbering of student records.

Writing TPI programs. The writing of the TPI programs also was done at the author's home prior to typing them into the computer. The HP 2115A at Wright-Patterson Air Force Base was used for typing and debugging the first few TPI questions. One of the most discouraging problems encountered during the initial TPI programming was the limited amount of storage that was available. The HP Educational BASIC manual suggested that programs of up to 250 statements could fit into the computer. In actuality, the TPI programs, which used every available
memory location, ranged from 162 to 177 statements with a mean of 166.

The TPI were originally designed to present 10 questions; however, because of the memory restrictions, the last two or three questions had to be discarded. Even with the discarding of these questions, the TPI programs had to be considerably modified to make them fit. This modification involved eliminating all documenting REM statements, some textual messages, and abbreviating words in others. This modification process was very tedious because the desire to eliminate as few characters as possible forced the continual trial execution of the program to determine if it fit. One technique that helped in this process was the temporary placement of a statement at the start of the program which transferred control to the last question. This eliminated having to "take" the TPI program each time a modification was made. Unfortunately, when the TPI programs were transferred to the HP 2114B at CCDS, even more stripping of textual characters had to be done. This was necessary because of the extra memory required for the calls to the student record programs.

A "start" tape such as the one used in check program development was again used for reducing the amount of redundant programming. This tape contained 113 BASIC statements and therefore resulted in an average of only 53 unique statements being necessary for each TPI program. One disadvantage of use of the start tape was that if an error was found in one of its statements, all TPI programs written up to that time had to be revised and repunched. Since each program required approximately 10 minutes to punch, these errors were quite consequential. This excessive punching time also caused the generation of TPI program
backup tapes to be discontinued after TPI program number 3. Certainly
the availability of a high-speed paper tape punch would have signifi-
cantly reduced the amount of time spent in program development. A
table listing TPI usage and programming data is given in Appendix C.
This appendix also includes a sample TPI printout.

H-P BASIC as a TPI language. Besides its lack of alphabetic character
input, H-P Educational BASIC has a few other disadvantages as a TPI
programming language. The formatting of numbers in text is limited to
the spacing characteristics of the PRINT command. This lack of for-
matting capability resulted in many unwanted spaces in the textual
material such as "( 2 , 3 )" instead of the more readable "(2,3)". The
"TAB" option of the PRINT statement was quite useful but had to be
discarded because it consumed too much memory. When a mistake was made
in typing a line of text, or when a word had to be abbreviated, the
whole line of text had to be retyped. The availability of a character
editor would have been extremely helpful in this effort.

Normally, when a BASIC program is first written, the programmer
uses statement numbers incremented by 10. This large increment allows
new statements to be inserted between previous statements without re-
quiring any retyping. However, the TPI programs were modified so ex-
tensively that at times there would exist three or more consecutive
statements with no gap to allow the insertion of any statements.
These consecutive statements then had to be renumbered and retyped.
The availability of a renumbering option which would automatically
resequence the statement numbers and restore the 10 increment would
have been helpful. This option could be called for in the same manner
4.6 Student Record Keeping

The permanent storing of student records in the computer required that six assembly language programs be written to update and retrieve data from the student data bank. These programs, to be referred to as "student record programs," also had to be permanently stored in the computer and protected from inadvertent student destruction. Since little adjustment could be made in the size of these programs, the quantity of student data that could be permanently stored was limited to the area of protected central memory that was not consumed by the student record programs.

The writing of the student record programs was done by the school's computer expert, Jeff Spain, with the assistance of the author. These programs were quite difficult to debug and were the sole cause of the course being delayed a week. However, once the programs were debugged, they didn't require any further debugging or modification for the duration of the course. In fact, except for a system error on the first and twelfth session of the course causing a reload of the BASIC interpreter, and an occasional clobbering of student data by an erroneous homework checking or TPI program, the record keeping procedure was relatively trouble-free.

The following paragraphs of this section will present more specific details of how the protected area of central memory was selected and allocated, and will include some of the programming considerations

1Many time-sharing systems do have the editing capabilities suggested in this section.
and problems in updating, querying, and safeguarding the student data bank.

Selecting location of student record programs and data bank. The primary consideration in establishing a student data bank was determining where the data and the record keeping programs should be stored. Because of the large number of short activities anticipated at the terminal, it was imperative that the sign-on procedure be as simple as possible. This decreased the desirability of storing the data bank and record programs on paper tape and requiring the student to load these tapes each time he signed-on.

Fortunately, after reviewing a diagram of the allocation of central memory during the use of BASIC, some unused cells (words) were found between the end of the BASIC interpreter and the first address of the user's BASIC program. This area was non-accessible from the BASIC interpreter and the user's program and, therefore, was ideally suited for storing student data and the record programs; however, it was not large enough. To increase this area, it was decided to reconfigure the BASIC interpreter to eliminate its instructions for decoding BASIC's matrix statements. This decision was made with trepidation because it eliminated the availability of the matrix operations for anyone using BASIC during the study, including students from other classes. The final arrangement of central memory is shown in Figure 10.
Central memory

- inaccessible from student programs
- area n normally occupied by matrix package

area available for reporting, homeworking checking and TPI programs

Fig. 10. Allocation of central memory

Arrangement of student data in data bank. Having established a reasonable amount of non-accessible permanent memory, it was necessary to determine how much of this area could be used for student data; it was decided that at most 400 16-bit words were available. To stay within this constraint, 90 words were allocated for quiz data, 180 words for homework data, and 90 words for TPI data. The arrangement of this data is illustrated in the author's letter to Jeff Spain in Appendix D.

The quiz data for an individual student consisted of five words. Each word was separated into four 4-bit parts with each part containing a quiz score. It was unnecessary to save the quiz number with the quiz score because the position of the 4-bit part was used to refer to the appropriate quiz. This same scheme, called "referencing by
position" was also used for placing data in the words used for homework and TPI records.

The homework data for an individual student consisted of ten words. Each word was also separated into four 4-bit parts with each part containing the number of times the student tried a program. If, and when, a student's program was determined to be correct, "8" was added to his number of tries. This resulted in a "1" being placed in the first bit of the 4-bit part.

The TPI data consisted of five words for each student with each word separated into two 8-bit parts, each part representing the use of a specific TPI program. The first two bits of the 8-bit part represented the number of times the student tried the TPI program and the last six bits contained the number of incorrect responses the student made on his most recent try of that TPI program. It would have been desirable to save the number of incorrect responses for each TPI program try but this would have consumed more memory than was available. The possibility of saving the average number of incorrect responses for all tries was also considered but was rejected because it would have added complexity to either the TPI programs or to the assembly language program to update the student's TPI record.

Updating and retrieval programs. Once the storage arrangement of the student data had been established, the next task was to specify the programs required to update the student records and retrieve data from them. These programs were to be written in H-P assembly language and referenced from the author's BASIC programs via a 'CALL' statement (Hewlett-Packard, 1970). Six programs were required; two each for quiz
data, homework data, and TPI data. One of the two programs for each activity retrieved student data; the other updated student records.

The programs handling quiz data retrieved and updated the eighteen scores for a quiz using just one CALL. This was done because all the student scores for a particular quiz were calculated during the same computer run. The other four programs retrieved and updated data using the student ID number and the number of the particular homework checking or TPI program being used. The programs handling homework data only transferred one parameter, "number of tries," between the data bank and the BASIC programs whereas the programs handling TPI data transferred two parameters, "number of tries" and "number of incorrect responses on latest try." The calling sequences, descriptions, and flow diagrams for these programs are also included in the author's letter to Jeff Spain in Appendix D.

Writing of student record programs. Since the student record programs had to be written in H-P assembly language with which the author was unfamiliar, Jeff Spain, a senior with considerable experience in H-P assembly language programming, volunteered to write the required programs. A detailed description of these programs was sent to Jeff approximately two weeks before the planned starting date of the course (see Appendix D).

Being an expert programmer, it only took Jeff a few hours to write the programs but, because of other school commitments, he didn't have a chance to check the programs on the computer until the afternoon of the day of the orientation lecture. This was only three days before
the planned starting day of the course. Naturally, the programs
didn't work the first time they were tried on the computer, and after
numerous debugging and correction runs, were still not running very
satisfactorily. At this point, Jeff decided to take the programs home
to look them over; this was Friday night.

The next afternoon, Mr. Klitz called to say that Jeff had spent
most of the night trying to find the errors in the programs but was not
successful and didn't have any more time to spend debugging that weekend because of a required term paper. This left the author and Mr.
Klitz with no alternative but to postpone the course for a week. The
author spent the rest of that weekend learning H-P assembly language.
The following Monday, the author went to CCDS to get Jeff's programs
and spent the next two evenings debugging them at home by using sample
data and simulating the computer instructions. Fortunately, this procedure found most of the errors and, after two more evenings debugging
via the console of the computer, the record programs were operational.

One decision during the debugging that considerably improved the
chances of success was the replacing of Jeff's subprograms for binary-
to-decimal and decimal-to-binary conversion with calls to similar

1It should be noted here that any correction of a assembly language statement requires well over an hour. This is due to having 1)
repunch the paper containing the assembly language program, 2) reload the assembler via the high-speed paper tape reader, 3) load the assembly language program for the assembly process, 4) load the assembly language program for generating a binary tape containing the assembled program, 5) load the assembly language program to print out the memory locations used for each statement and the data, and 6) load the binary tape for subsequent execution. This procedure was not made any easier by the oversensitivity of the paper tape reader which caused many programs to have to be reloaded several times and, in some cases, re-punched.
routines in the BASIC interpreter. Although Jeff's subprograms were quite elegant, it was felt that much time could be saved by taking advantage of the assured reliability of the Hewlett-Packard routines; in fact, Jeff suggested this replacement and wrote the instructions to implement it. Rather ironically, this replacement could not have occurred if CCDS had not received an unsolicited assembly language listing of the BASIC interpreter the day before the replacement was made.

Safeguarding student records. Although the student data bank was fairly well protected from errors in BASIC programs, it was still vulnerable to a student inadvertently loading the Fortran compiler or H-P assembler into the computer and destroying the student records. An equally disastrous event would have been the reloading of the BASIC interpreter due to a system error and neglecting to delete the matrix package. To circumvent these problems, two additional assembly language programs were written; one for dumping the student records onto paper tape and one for reading the records from paper tape back into central memory. The dumping program was used each morning to guarantee a loss of no more than one day's data. These dump and load programs were also used by the more advanced programmers when they wanted to program in Fortran or H-P assembly language.

4.7 Daily Operating Procedures

Each classroom session began with a ten-minute quiz which was discussed after the quiz cards were collected. Occasionally the quiz took longer than ten minutes because a student forgot his quiz cards and had to be instructed in how to mark them with a pencil. Some
class time was allocated during the first few lectures for running programs on the computer but, as the course progressed in difficulty, lecture time became a premium and this practice was eliminated. Eventually, Mr. Klitz resorted to an overhead projector with predrawn diagrams to increase the amount of information presented during a lecture.

There was considerable pressure to cover the required material for each lecture because the following day's quiz had questions pertaining to it.

The lack of student operators caused the students to have to load their own programs. This inefficiency, combined with the lack of scheduling procedures, resulted in the computer room becoming quite congested at the beginning of the course. However, as the course progressed and some of the students got discouraged, the first-come, first-served arrangement seemed to be adequate for those students who continued to use the computer. Nonetheless, in retrospect, it seems that a scheduling procedure should have been used. It would have decreased student time wasted waiting on the computer and would have helped reduce student anxiety caused by the crowded environment.

4.8 Tests and Evaluation

Pretest. The pretest was generated by the author and administered by Mr. Klitz during one 45 minute class session. The test was separated into two parts: analytic geometry and BASIC. The first part consisted of ten questions taken from sections of the textbook to be covered during the first seven lectures and the second part consisted of seven questions concerned with BASIC language syntax, flowchart usage, storage allocation, and BASIC program writing. Appendix E contains a
copy of the pretest, a listing of the sections of the text that the
questions were taken from, and an item analysis of the questions.
None of the students completed the test.

Final Exam. The major criteria in selecting a final exam was to iden-
tify a level of accomplishment recognized as acceptable for credit by
a college or university. Ideally this examination would be a final
exam from an institution that used Contemporary Analytic Geometry as
a textbook for an analytic geometry course. The pass-fail criteria of
this institution could then be used to determine how many of the CCDS
students would have passed the college course. To determine if such
an ideal examination existed locally, Mr. Klitz called McGraw-Hill,
the publisher of the text, to ascertain if any institutions in the
tri-state area (Ohio, Kentucky, Indiana) were using the text. The
person contacted at McGraw-Hill said that she would search their
records but, after a few days, notified Mr. Klitz that there were no
other local users of the text.

The next alternative was to contact a local college or university
that was teaching analytic geometry and to use questions from their
final exam with the passing criteria adjusted for the number of items
chosen. From previous conversations, the author was aware that Mr.
Tom Bruggeman of Xavier University, Cincinnati was teaching an ana-
lytic geometry course. However, Mr. Bruggeman discouraged the use of
any of his exams because of their emphasis on the vector approach and,
instead, offered the teacher's guide to a programmed text in analytic
geometry. This guide contained four equivalent exams for each of the
two parts of the programmed course and a table of statistics for
comparison of the test scores to established norms. The two parts of the programmed course were titled "The Line" and "The Conics."

To determine the time requirements of this exam, the author administered Part I and Part II of the test to himself. Part I required 16 minutes to complete and Part II took 40 minutes. Since the available class time for the final exam was only 45 minutes, the projected time for taking the exam under consideration was too long, especially if it was assumed that the students would require more time than the author. To solve this timing problem, it was decided to eliminate the very time-consuming last problem of Part II. This seemed to have solved the problem because most students were able to complete the exam when administered.

Although there were no timing problems, the students were not very pleased with the final exam. They felt that it inadequately represented the content of the course because of its overemphasis of conics and lack of questions on set theory, polar coordinates, and solid analytic geometry. A copy of the final exam and an item analysis is available in Appendix E.

Student evaluation. Except for occasional student complaints to Mr. Klitz about incomplete lectures, "computer hogs," and faulty check programs, most of the student evaluation occurred during the class session following the final exam. This entire 45 minute session was dedicated to completing the student questionnaire. The dedication of a class session for evaluation was felt necessary to allow the students to elaborate on the questions asked. It is quite doubtful that the lucid comments received would have been generated if the students
were given the questionnaires to fill out at home or during the waning moments of a normal class session. A summary of these comments is included in the following chapter; the original comments are presented in Appendix F.

**Teacher evaluation.** Mr. Klitz kept a daily record of his activities and of his thoughts toward the course. Some of his comments have already been quoted in previous sections of this chapter and others will be quoted in the following chapter. These comments were also used to identify problems for the author's consideration during the execution of the course. In addition to his anecdotal remarks, Mr. Klitz was asked to complete a modified student questionnaire.

**Author's diary.** The author originally kept a record of time spent on significant events such as generation of quizzes, homework assignments, instruction sheets, quiz cards, checks programs, TPI programs, etc. However, as the programming effort became increasingly time-consuming, it was decided to record the time spent on each individual programming step. These steps included writing the program, debugging, making copies, generating correct solutions, etc. This record allowed some indication of the length of time necessary to construct each individual program and will be referred to in Chapter Six as useful data for estimating the feasible number and size of programs to be included in any future implementations.

This completes the description of the implementation of the plans outlined in Chapter Three. The following chapter will report the data collected during this implementation.
CHAPTER FIVE
REPORT OF DATA

5.1 Introduction

Pretest results, final exam results, and data for the computerized activities (quizzes, programming and TPI) will be presented in the following sections of this chapter. In addition, student and teacher evaluation data, student homework times, and student background data was also collected. Summarizations of this data will be included in those sections for which it is most appropriate. The chapter will conclude with a correlational analysis of course components and a description of some attempts at measuring achievement.

A summarization of the data collected during the course is presented in Table 4. A very significant variable, average student homework time, is reported in this summary as 52 minutes. However, because the distribution is somewhat skewed due to the ambitious programmers, its median of approximately 42 minutes should also be considered for comparison to time spent on other courses. The standard deviation of 26 minutes indicates a rather large variation of time spent by the students, again probably due to the ambitious programmers.

With regards to student evaluation of the course, one student felt he learned more, eight felt they learned less, and eight felt there was no difference between a computer course and a normal classroom approach. Concerning student morale, nine students indicated that they enjoyed
### Table 4

Overall student summary

<table>
<thead>
<tr>
<th>Student Identification Number</th>
<th>Pretest</th>
<th>Quiz Average</th>
<th>TPI Programs Tried</th>
<th>Final Exam</th>
<th>Average Time Spent on HW</th>
<th>Attitude Index</th>
<th>Course Grade Average</th>
<th>All School Grade Average</th>
<th>IQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>61</td>
<td>3.7</td>
<td>0</td>
<td>0</td>
<td>88.5</td>
<td>41</td>
<td>+1</td>
<td>71</td>
<td>75</td>
</tr>
<tr>
<td>2</td>
<td>80</td>
<td>5.7</td>
<td>22</td>
<td>19</td>
<td>0</td>
<td>140</td>
<td>76</td>
<td>+2</td>
<td>91</td>
</tr>
<tr>
<td>3</td>
<td>80</td>
<td>5.5</td>
<td>32</td>
<td>29</td>
<td>9</td>
<td>140.5</td>
<td>118</td>
<td>+3</td>
<td>91</td>
</tr>
<tr>
<td>4</td>
<td>60</td>
<td>4.2</td>
<td>4</td>
<td>1</td>
<td>81</td>
<td>24</td>
<td>-1</td>
<td>75</td>
<td>77</td>
</tr>
<tr>
<td>5</td>
<td>63</td>
<td>4.0</td>
<td>9</td>
<td>8</td>
<td>105</td>
<td>33</td>
<td>+1</td>
<td>72</td>
<td>74</td>
</tr>
<tr>
<td>6</td>
<td>53</td>
<td>5.0</td>
<td>26</td>
<td>20</td>
<td>95.5</td>
<td>79</td>
<td>+2</td>
<td>85</td>
<td>84</td>
</tr>
<tr>
<td>7</td>
<td>73</td>
<td>4.7</td>
<td>17</td>
<td>15</td>
<td>95.5</td>
<td>50</td>
<td>-1</td>
<td>82</td>
<td>80</td>
</tr>
<tr>
<td>8</td>
<td>44</td>
<td>3.7</td>
<td>0</td>
<td>0</td>
<td>67</td>
<td>28</td>
<td>-3</td>
<td>72</td>
<td>78</td>
</tr>
<tr>
<td>9</td>
<td>53</td>
<td>4.2</td>
<td>21</td>
<td>19</td>
<td>59.5</td>
<td>71</td>
<td>-1</td>
<td>79</td>
<td>75</td>
</tr>
<tr>
<td>10</td>
<td>50</td>
<td>4.3</td>
<td>1</td>
<td>1</td>
<td>123</td>
<td>43</td>
<td>-2</td>
<td>74</td>
<td>76</td>
</tr>
<tr>
<td>11</td>
<td>55</td>
<td>5.5</td>
<td>6</td>
<td>4</td>
<td>119</td>
<td>41</td>
<td>+1</td>
<td>87</td>
<td>84</td>
</tr>
<tr>
<td>12</td>
<td>65</td>
<td>4.4</td>
<td>1</td>
<td>0</td>
<td>102.5</td>
<td>28</td>
<td>-2</td>
<td>86</td>
<td>81</td>
</tr>
<tr>
<td>13</td>
<td>59</td>
<td>4.3</td>
<td>4</td>
<td>3</td>
<td>95.5</td>
<td>70</td>
<td>0</td>
<td>79</td>
<td>81</td>
</tr>
<tr>
<td>14</td>
<td>48</td>
<td>4.2</td>
<td>0</td>
<td>0</td>
<td>74.5</td>
<td>46</td>
<td>+2</td>
<td>73</td>
<td>77</td>
</tr>
<tr>
<td>15</td>
<td>31</td>
<td>4.6</td>
<td>9</td>
<td>7</td>
<td>98</td>
<td>35</td>
<td>0</td>
<td>80</td>
<td>77</td>
</tr>
<tr>
<td>16</td>
<td>52</td>
<td>3.2</td>
<td>0</td>
<td>0</td>
<td>84</td>
<td>33</td>
<td>-2</td>
<td>75</td>
<td>78</td>
</tr>
<tr>
<td>17</td>
<td>77</td>
<td>3.8</td>
<td>21</td>
<td>18</td>
<td>93.5</td>
<td>86</td>
<td>0</td>
<td>85</td>
<td>81</td>
</tr>
<tr>
<td>18</td>
<td>67</td>
<td>4.5</td>
<td>0</td>
<td>0</td>
<td>107.5</td>
<td>31</td>
<td>0</td>
<td>80</td>
<td>78</td>
</tr>
<tr>
<td>Maximum</td>
<td>100</td>
<td>100</td>
<td>7</td>
<td>38</td>
<td>38</td>
<td>162</td>
<td>+3</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td><em>Class Mean</em></td>
<td>60</td>
<td>24</td>
<td>4.4</td>
<td>9.6</td>
<td>8.0</td>
<td>2.6</td>
<td>98</td>
<td>0</td>
<td>80</td>
</tr>
<tr>
<td>Std Dev</td>
<td>13</td>
<td>13</td>
<td>.7</td>
<td>10.6</td>
<td>9.4</td>
<td>2.9</td>
<td>22</td>
<td>26</td>
<td>1.7</td>
</tr>
</tbody>
</table>

Maximum: 100
Class Mean: 60
Std Dev: 13
the course, three indicated they didn't, and four were indifferent.
The course content was almost unanimously (16 out of 17) stated to be
not too difficult if taught at a slower pace. Considering future use
of the computer, six students stated they would like their future math
courses taught with computer integration, five stated they would not,
and seven were indifferent. With regards to which activities were most
contributory to their learning, eight students indicated classroom lec-
tures, eight chose the textbook, two chose programming, and one chose
the quizzes.¹ The mean percentage effectiveness of the activities, as
indicated by the students, was 35% for lectures, 39% for textbook, 9%
for programming, 12% for quizzes, and 5% for TPI. A complete tabula-
tion of the evaluation sheets is available in Appendix F.

Most of the students included extensive comments on their evalu-
ation sheets; a compilation of these comments is also presented in
Appendix F. A short summary of these comments, excluding those re-
lated to the tests and the individual computerized activities, is pre-
sented below:

Computer availability: Not enough time available for all students
to use the computer. A small minority monopolized the time that
was available.

Course length: The course was much too short for the amount of
material covered and student participation expected.

Individuality: Some students stated they enjoyed the independence
of the approach but one student felt it allowed him to be "rather
negligent the entire time."

Integration of computer activities: Most of the students felt
that the computer was interesting but detracted from their

¹This total more than 18 because three students indicated that
the lectures and textbook were equally effective.
learning of analytic geometry. They recommended less emphasis on
the computer and more on the mathematics.

With regards to the teacher evaluation, Mr. Klitz stated that he
felt the students learned more and enjoyed the course. His distribu-
tion of the percentage of activities contributing to the students
learning was 20% for lectures, 20% textbook, 30% programming, 20%
quizzes, and 10% TPI. He concluded his evaluation with the remarks:

My overall comment would be satisfaction with the approach. I
think it would have been more enjoyable if we had not been under
the 4 week time period, and had the course not been squeezed in
at the end of the year. However, these were mechanical prob-
lems and not necessarily related to the computerized approach.

Mr. Klitz's complete evaluation sheet is also available in Appendix F.

5.2 Pretest

The pretest was separated into two parts: analytical geometry and
BASIC programming. Each part had a maximum score of 100. The analyti-
cal geometry portion assessed the student's knowledge of the first 2 1/2
chapters of the text. The mean score on this portion was 60 with a
range of 31-80 and a standard deviation of 13. The average percentage
correct for the questions associated with the individual chapters of
the text was 60% for Chapter 1, 52% for Chapter 2, and 67% for Chapter
3.

The mean score on the BASIC portion of the test was 24 with a
range of 2-50 and a standard deviation of 13. Part of the cause of
such poor results was due to the time constraints of the test; however,
these poor results also revealed a significant lack of knowledge of
BASIC for some of the students. A complete summary of the pretest re-
results is available in Appendix E.
5.3 **Quizzes**

The average quiz score for all students for all quizzes was 4.4. Since the maximum score on each quiz was 7, this average converts to 63%. The students quiz averages ranged from 3.2 (43%) to 5.7 (81%) with a standard deviation of .7 (10%). The class average on individual quizzes ranged from 2.8 (40%) on quiz no. 18 to 5.8 (83%) on quiz no. 5. A complete summary of the student quiz scores is available in Appendix A.

With regard to student opinion on the difficulty of the quizzes, only one student stated they were too difficult whereas 13 students and Mr. Klitz stated they were not. In addition, four students indicated that the quizzes would not have been too difficult if the course was taught at a slower pace. Comments on the quizzes were quite varied with some students praising them and others criticizing them for usurping class time and being "rediculously piciune."

5.4 **Programming**

The average number of programs tried per student was 9.6 with a range of 0 to 32 and a standard deviation of 10.6. The average number of programs completed was 8.0 with a range of 0 to 29 and a standard deviation of 9.4. Five students didn't try any programs and four others tried less than five problems. This distribution of problems tried and completed by student is presented in Figure 11 and by session number in Figure 12. The number of students programming each day is presented in Figure 13 and the number of times each homework program was tried and completed is presented in Figure 14.

A total of 173 problems were attempted with 144 being certified.
Fig. 11 Programs per student

Fig. 12 Programs per session
Fig. 13 Number of students programming each day

Fig. 14 Tries per homework problem
as correct by the check programs. Altogether, there were 515 individual runs through the check programs with an average of 3.4 tries per problem for unsuccessful programs and 2.9 tries for successful programs. Programming results are summarized in Appendix B. A least squares regression line was used to relate total homework time to programs tried. The equation of this line was \( t=38.6p + 536 \) which estimates that a student spent 536 minutes on homework (30 minutes per night), regardless of his programming activity, and an additional 38.6 minutes for each program attempted. The standard error of estimate was 222 minutes (12 minutes per night).

On the evaluation questionnaire, the students were asked if they felt the required programs were too difficult. Only two students responded "yes" while four responded "no" and eight responded "no, if I knew more programming." Almost every response, however, was qualified with some comment. The reader is encouraged to review these comments in their original form in Appendix F; however, an attempt has been made to condense them into the following statements:

- All students should have been properly prepared in programming
- More time should be allowed for the programs
- Students should have an unlimited number of tries
- The check programs should be more carefully checked themselves
- The programs didn't teach very much mathematics.

Mr. Klitz's felt the required programs were too difficult but qualified his remarks with the statement: "A difficult question - For the able programmers, no - But, for many of the students they were too difficult."
5.5 Terminal Presented Instruction

The average number of TPI programs tried per student was 2.6 with a range of 0 to 9 and a standard deviation of 2.9. Comparing the total number of wrong responses (80) to the number of programs tried (44) resulted in an average of 1.6 wrong responses per program try. Although students were allowed to try the same TPI program up to three times, only two students tried any TPI program more than once. The distribution of TPI programs tried by student is presented in Figure 15 and by session number in Figure 16. The number of students using TPI each day is presented in Figure 17 and the number of tries per TPI program is illustrated in Figure 18. A summary of TPI usage is available in Appendix C.

When asked if the TPI programs were useful, twelve students and Mr. Klitz responded "Yes," three responded "No," and three responded "Didn't try any." The consensus of the student comments concerning the TPI programs was 1) they were enjoyable, 2) they just skimmed the surface, 3) more computer response should be given when a question is missed, and 4) the abbreviations and contractions of words made the reading rather difficult. Mr. Klitz commented that he "would have liked to see them used more."

5.6 Final Exam

The maximum score on the final exam was 162. The student average was 98 (60%) with a range of 59.5 (37%) to 140.5 (87%) and a standard deviation of 22 (13%). (It seems worth noting that the mean of the analytical geometry pretest, the quiz average, and final exam mean were 60%, 63% and 60% respectively, and their standard deviations were
Fig. 15 TPI programs tried per student

Fig. 16 TPI programs tried per session
Fig. 17 Number of students using TPI each day

Fig. 18 Tries per TPI program
The final exam was separated into two parts, "The Line," worth 100 points, and "The Conics," worth 62 points. The student average of these two parts was 60 (60%) for the first part and 38 (61%) for the second part.

When comparing the CCDS class average of 60 on Part I to the mean of 87.6 for 128 freshman students taking Part I at Depauw University in 1964 (Davis, 1967, p. 3), the CCDS results do not look very promising. However, the Depauw students spent four weeks (four 1-hour classes per week) studying the material related to Part I whereas the CCDS students only spent eight 45-minute class sessions on the same material. In addition, the Depauw students had an hour to take the exam whereas the CCDS students had only 45 minutes for both Part I and three questions of Part II.

The results on Part II also reflect this disparity of time spent. The mean for 123 Depauw freshman students taking all of Part II in 1965 was 81.3. This is considerably higher than the 60 percent CCDS average on the first three questions. However, the Depauw students had 16 hours of lecture and one hour to take the exam compared to 4 1/2 hours of lecture and part of the 45 minute exam period for the CCDS students. To further discourage any hopes for comparison, one should not ignore the rather high IQ average of the CCDS students and the more than three year age difference between the groups.

When asked if the final exam was representative of what they had learned, no student responded "Yes," seven students responded

---

¹Not all five questions of Part II were used because of time constraints and not because the material hadn't been covered.
"Somewhat," and nine students responded "No." The student comments were very critical of the exam, especially with its emphasis on conics. Psychologically, they seemed to have felt denied of their right to have their knowledge tested by a representative exam. This may have influenced their negative response to the "have you learned more" question on the evaluation form which was completed the day following the final exam.

5.7 Correlation Analysis

The program BMD02D - Correlation with Transgeneration of the BMD Biomedical Computer Programs (Dixon, 1971) was used to calculate correlations between the variables represented in Table 4. The results of this program are displayed in Table 5. These results seem to indicate that very few of the correlations between the variables were positive (p < .05). Those that were significant could have been anticipated except, possibly, for the positive correlation between quiz average and geometry grade.

An attempt was made to relate homework problems to posttest questions to determine if there was any relationship between having programmed a problem and correctly answering a similar problem on an exam. The results for the seven programs that satisfied this criteria are presented in Table 6. These results indicate that for all programs, except number 22, the average final exam item score for those programming the problems was higher than the class average. An equally interesting observation is that the test item scores were almost always below the class average for those students who had unsuccessfully tried to program the related problem.
<table>
<thead>
<tr>
<th>Measure</th>
<th>I.Q.</th>
<th>All School Grade</th>
<th>Geometry Grade</th>
<th>Attitude Index</th>
<th>Average HW Time</th>
<th>Final Exam</th>
<th>TPI Tried</th>
<th>Programs Completed</th>
<th>Programs Tried</th>
<th>Quiz Average</th>
<th>Pretest - BASIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pretest - Analytical Geometry</td>
<td>.49</td>
<td>.50</td>
<td>.55</td>
<td>.35</td>
<td>.51</td>
<td>.51</td>
<td>.51</td>
<td>.48</td>
<td>.34</td>
<td>.53</td>
<td></td>
</tr>
<tr>
<td>Pretest - BASIC</td>
<td>.65</td>
<td>.39</td>
<td>.41</td>
<td>.09</td>
<td>.30</td>
<td>.24</td>
<td>.13</td>
<td>.42</td>
<td>.40</td>
<td>.20</td>
<td></td>
</tr>
<tr>
<td>Quiz Average</td>
<td>.56</td>
<td>.72</td>
<td>.79</td>
<td>.60</td>
<td>.47</td>
<td>.71</td>
<td>.33</td>
<td>.54</td>
<td>.56</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Programs Tried</td>
<td>.56</td>
<td>.59</td>
<td>.69</td>
<td>.53</td>
<td>.86</td>
<td>.34</td>
<td>.53</td>
<td>.99</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Programs Completed</td>
<td>.56</td>
<td>.58</td>
<td>.67</td>
<td>.52</td>
<td>.87</td>
<td>.34</td>
<td>.52</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TPI Tried</td>
<td>.19</td>
<td>.25</td>
<td>.25</td>
<td>.47</td>
<td>.44</td>
<td>.09</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Final Exam</td>
<td>.27</td>
<td>.62</td>
<td>.63</td>
<td>.47</td>
<td>.38</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average Homework Time</td>
<td>.47</td>
<td>.64</td>
<td>.63</td>
<td>.59</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attitude Index</td>
<td>.27</td>
<td>.43</td>
<td>.42</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Geometry Grade</td>
<td>.75</td>
<td>.32</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>All School Grade</td>
<td>.75</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* p < .05
Table 6
Comparison of program completion to final exam scores

<table>
<thead>
<tr>
<th>Program for Problem Number</th>
<th>Associated Final Exam Questions</th>
<th>Possible Score</th>
<th>Class Average</th>
<th>Students Completing Program Number</th>
<th>Average Score</th>
<th>Students Trying but not Completing Program Number</th>
<th>Average Score</th>
<th>Correlation Between Number of Tries and Score on Final Exam Questions</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>LINE 2,3,4, &amp; 5</td>
<td>12</td>
<td>8.3</td>
<td>9</td>
<td>10.3</td>
<td>2</td>
<td>6.0</td>
<td>- .28</td>
</tr>
<tr>
<td>11</td>
<td>CONICS 5 &amp; 6</td>
<td>4</td>
<td>3.6</td>
<td>3</td>
<td>3.7</td>
<td>0</td>
<td></td>
<td>.98</td>
</tr>
<tr>
<td>16</td>
<td>LINE 11</td>
<td>12.5</td>
<td>9.0</td>
<td>5</td>
<td>12.5</td>
<td>1</td>
<td>0</td>
<td>undefined</td>
</tr>
<tr>
<td>16</td>
<td>LINE 10</td>
<td>12.5</td>
<td>12.5</td>
<td>6</td>
<td>12.5</td>
<td>1</td>
<td>12.5</td>
<td>undefined</td>
</tr>
<tr>
<td>22</td>
<td>CONICS 31 &amp; 32</td>
<td>12</td>
<td>3.0</td>
<td>5</td>
<td>2.4</td>
<td>0</td>
<td></td>
<td>-.40</td>
</tr>
<tr>
<td>23</td>
<td>CONICS 17, 26 &amp; 27</td>
<td>5</td>
<td>2.4</td>
<td>4</td>
<td>2.5</td>
<td>1</td>
<td>0</td>
<td>-.58</td>
</tr>
<tr>
<td>28</td>
<td>CONICS 1 &amp; 16</td>
<td>6</td>
<td>4.8</td>
<td>2</td>
<td>6.0</td>
<td>0</td>
<td></td>
<td>undefined</td>
</tr>
</tbody>
</table>
A similar analysis was made relating TPI programs to final exam questions (see Table 7).

Table 7
Comparison of TPI program usage to final exam scores

<table>
<thead>
<tr>
<th>TPI Program Number</th>
<th>Associated Final Exam Questions</th>
<th>Possible Score</th>
<th>Class Average</th>
<th>Students Taking TPI Program No.</th>
<th>Average Score</th>
<th>Correlation Between No. Wrong on TPI Program &amp; Score on Exam Questions</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>LINE 2, 3, 4, 5, 7 CONICS 1, 5, 6</td>
<td>22</td>
<td>17.6</td>
<td>9</td>
<td>17.4</td>
<td>-.19</td>
</tr>
<tr>
<td>4</td>
<td>LINE 10, 11, 12</td>
<td>37.5</td>
<td>31.3</td>
<td>5</td>
<td>30.0</td>
<td>-.53</td>
</tr>
<tr>
<td>6</td>
<td>CONICS 31, 32</td>
<td>12</td>
<td>3.0</td>
<td>3</td>
<td>6.0</td>
<td>.87</td>
</tr>
<tr>
<td>7</td>
<td>CONICS 1, 16, 17, 19, 20, 26, 27</td>
<td>17</td>
<td>9.4</td>
<td>4</td>
<td>10.0</td>
<td>-.71</td>
</tr>
</tbody>
</table>

These results seem to indicate that interacting with a TPI program had very little influence on the student's outcome on the final exam. Not enough subjects were available to place any emphasis on the correlations between the number of questions missed when interacting with a TPI program and the students score on the final exam questions. However, this statistic could be quite useful in evaluating future TPI programs because, if future TPI programs are to be effective, the correlation should be very close to zero, i.e., a student should
finish the TPI program with equivalent knowledge of the subject matter presented regardless of the route taken.

5.8 Achievement

The inadequacies of the pretest and final exam have been sufficiently discussed to explain the inability to measure overall class achievement. However, an analysis was made of the results on four comparable items on the pretest and final exam (see Table 8).

Table 8
Comparison of results on similar pretest and final exam questions

<table>
<thead>
<tr>
<th>Pretest Question No.</th>
<th>Class Average on Pretest Question</th>
<th>Final Exam Question No.</th>
<th>Class Average on Final Exam Question</th>
</tr>
</thead>
<tbody>
<tr>
<td>6a</td>
<td>83%</td>
<td>Line 7</td>
<td>100%</td>
</tr>
<tr>
<td>6c</td>
<td>22%</td>
<td>LINE 11</td>
<td>72%</td>
</tr>
<tr>
<td>10a</td>
<td>89%</td>
<td>CONICS 5</td>
<td>94%</td>
</tr>
<tr>
<td>10b</td>
<td>78%</td>
<td>CONICS 6</td>
<td>78%</td>
</tr>
</tbody>
</table>

These results can only be interpreted as miniscule evidence that the computerized approach did not cause the students to regress in their knowledge of the subject matter assessed by these questions.

This completes the report of the data collected during the study. The following chapter will describe observations regarding the study objectives defined in Chapter 1 and will describe some recommended follow-on activities.
CHAPTER SIX
SUMMARIZATION AND FUTURE RESEARCH

6.1 Introduction

Sections 6.2 and 6.3 of this concluding chapter will discuss the limitations and observations of the study. For those readers who find merit in the integrated, computerized approach to analytic geometry and wish to include all or part of the design in a future mathematics course, recommendations for future classroom implementation are described in Section 6.4.

Although one of the goals of a developmental effort is to establish the existence of an alternative way of doing things, an equally important goal is to "debug" the new alternative so that it can be properly compared to the existing alternatives. Hopefully, as a result of this study, the integrated, computerized approach is now ready to be compared to a conventional non-computerized approach. For those researchers who concur with this assumption, Section 6.5 of this chapter contains suggestions for future comparative evaluations of the computerized teaching model described herein.

The chapter will conclude with some suggestions for future research and development by teachers, doctoral candidates, professional societies, and computer firms. These suggestions have evolved from the author's participation in the study and, although they are not all direct consequences of this study, they do represent the results of the author's thoughtful, retrospective analysis of some of the
problems facing persons attempting to integrate the computer into mathematics teaching.

6.2 Limitations

To be able to conclude from an experiment that "Doing A will significantly influence B, assuming C" is a very rewarding experience for a researcher. Unfortunately this type of statement is difficult to make in most developmental educational research because of the lack of effective controls. The study presented herein is no exception. The sample size was small; there was no control group; and the course was taught under a time constraint not generally encountered in the real world. These were comparative limitations; there were also developmental limitations related to computer capability, and implementation limitations related to student background and motivation. These limitations, and limitations in the testing procedure, will be described in the following paragraphs.

Students. Because of the developmental nature of the program, an agreement was made with the students to not enter their success or failure into the school records and to not let their efforts influence their current geometry grade. This lack of pressure was admitted by a few students as a major contributor to their poor showing. However, the average amount of homework time spent per student per night was 52 minutes which is probably more than the average homework time for their other courses, or the just-completed geometry course. Consequently, it is somewhat debatable whether the student's freedom and anonymity was really a limitation.

Probably the most significant limitation in testing the value of programming was the lack of student programming skills. Although the
students had an introductory course and a few days review before the first analytic geometry class, some of the students never did learn to program to any degree of proficiency. However, even if all students had been able to program well, the resulting activity might have exaggerated another inherent limitation; that of lack of available computer time.

Computer. Considering the student class schedules, there just wasn't enough terminal time available for all students to debug their programs and interact with the TPI programs. A few other limitations with regards to the computer were 1) the lack of a high-speed paper tape punch to reduce the check program and TPI development time, 2) the necessity for each student to have to load the desired check programs and TPI from paper tape, 3) the limited amount of protected memory for student records, and 4) the difficulty in marking the BASIC programming cards.

Testing. Whether it should be considered a limitation or just poor planning, the testing portion of the study was grossly inadequate. The analytic geometry section of the pretest did not adequately relate to the material to be covered and did not test the limits of the student's knowledge. In addition, no controls were placed on the division of time between the analytic geometry and BASIC sections. Actually, since BASIC was not the subject being taught, it probably should not have been part of the pretest unless, of course, improvement in BASIC programming was to be measured as part of the posttesting.

The final exam was equally as bad! According to the students, it inadequately represented what they had learned. There was also very little overlap of comparable questions between the pretest, quizzes,
and final exam and therefore it was impossible to measure growth of achievement. A better designed testing program might have been able to detect improvements in problem solving capabilities, programming skills, and analytic geometry achievement that were caused by specific activities or interactions of activities. A later section of this chapter, titled "Suggestions for Comparative Evaluations," will describe a revised testing procedure that will hopefully eliminate some of these inadequacies.

6.3 Observations

This section will first address the investigations itemized in Section 1.7, "Significance of the Study," and then will discuss additional observations made during the study.

Existence. The existence of an integrated, computerized secondary school analytic geometry course was established. Although not all possible computer activities discussed in Section 1.4 were considered for this study, the integrating procedures used were general enough that they could be rather easily modified to allow integration of additional activities; assuming, of course, that sufficient computer power is available to implement these activities.

One very important observation in the organization of the course was the tendency to keep the computer busy regardless of the value of the activity. Future research on each of the course activities, along with more effective record keeping programs and prescription algorithms, should help discourage this tendency and result in a more efficient use of the computer resource.

Use of minicomputer. At times, especially during the writing of the
assembly language programs and the punching and rewinding of paper
tape, the author wished that a terminal to a large time-sharing com-
puter had been available. Time sharing would have eliminated the need
for assembly language programs and paper tape storage, and would have
allowed more comprehensive record keeping. However, it probably
would have cost more because of the large amount of auxiliary memory
necessary for storing student records and support programs, and because
of the constant, around-the-clock usage of the computer for develop-
ment and implementation.¹

The use of a minicomputer was also advantageous over most time-
sharing systems because of the availability of card input for quiz
grading and homework checking. These cards made possible the collection
of data at its source, i.e., in the classroom during a quiz or at home
while writing a program, and allowed the quick entry of this data into
the computer via the high-speed card reader. Another advantage of the
minicomputer was its exceptional reliability; the one or two machine
malfunctions that did occur were early in the course and were immed-
imately remedied.

The only unreliable peripheral equipment was the high-speed paper
tape reader. During the early part of the course this device was quite
sensitive to the dust that occasionally accumulated on the paper tapes

¹Although many schools do have 24-hour, unlimited usage, time-
sharing arrangements for a fixed fee, the future continuance of such
a policy on the part of the time-sharing suppliers is questionable.
As more schools and commercial firms begin to use time sharing, and as
time sharing tends to become a computer utility, the competitive mar-
ket may force all time-sharing suppliers, including universities, to
adopt a "pay for what you use" pricing policy.
and resulted in many tapes having to be recopied after just a few times through the reader. However, an adjustment of its sensitivity by a Hewlett-Packard representative helped remedy this problem. The high-speed tape reader was a vital component in the implementation of this course because the use of paper tape for program storage would not have been feasible without it.

The required use of assembly language for student record keeping was certainly a disadvantage\(^1\) of using a minicomputer; however, the alternative of using paper tape storage for student records was even less desirable. One recent low-cost alternative to paper tape storage that could also decrease the need for assembly language programming is the availability of tape cassettes for storing data\(^2\) and programs.

**Feasibility of teacher development.** The development of course materials would have been much too time-consuming for any teacher who tried to conduct the course without any assistance. The programming itself was not particularly difficult, except the assembly language portions, but the volume of programs and the requirement for "perfection" would have just overwhelmed a teacher with

---

\(^1\)Although the use of assembly language was a disadvantage for course development, its use was actually a very stimulating experience for the author. Although the author had been involved with computers for over eight years, he really never had a firm understanding of how a computer operates until he used the computer console for debugging the assembly language programs. In addition, it was the first time that he felt he was really in control of the computer. This experience is what minicomputer advocates describe as its "hands-on" advantage.

\(^2\)Of course these cassettes can also be used with time sharing which further complicates the comparison of time sharing to minicomputers.
responsibility for three or four other classes as well as the computerized class. Some of the development effort could have been reduced by 1) having a high-speed paper tape punch and additional computer memory, 2) lengthening the course and eliminating some of the homework programs, and 3) obtaining student assistance to help in menial tasks such as making duplicate paper tapes of programs and running the quiz deck. Probably the most feasible solution to the teacher time requirement for a course of this type, assuming the teacher is to do the development, is to have the teacher spend the previous summer developing the course materials. Other alternatives would be to have the material generated by the school's computer science clubs, supplied by computer vendor, or purchased from a software firm.

Feasibility of textbook sections as organizing units. Although textbook sections were used for organizing the course schedule, the actual unit used for integrating the quizzes, homework, and TPI programs was the "session number." The use of this less definitive unit was caused by the memory limitation for student records. This restriction allowed only total quiz scores and TPI questions missed to be saved rather than student responses on individual quiz items or TPI questions.

Assuming individual quiz and TPI question responses could have been saved, it is quite conceivable that a revised prescriptive algo-
algorithm could have been implemented that would have related quiz questions, TPI questions, and homework problems to textbook sections. Such an algorithm could have identified the textbook sections causing student difficulties and recommended activities for those section.

With regard to the student testing, it seems that much of the
inadequacy of the pretest and final exam was due to the lack of any unit for coordinating the exam questions with the course activities.

In summary, although textbook sections were adequate as an "organizing unit" for the course, they were not feasible as an "integrating" unit. However, the use of textbook sections as an integrating unit seems quite attractive and therefore should be considered for future implementations, even at the cost of reorganizing the student records or purchasing additional memory.

Prepunched quiz cards. Despite the prepunching, errors in marking the quiz cards were far too numerous and quite annoying. A better designed card would have helped, as would the assistance of a student data editor.

Feasibility of short TPI programs and BASIC as a language for implementing them. The TPI programs were liked by the teacher and the students and might have been used more frequently if the students had had a preview session before the course started. Although the short TPI programs were not used extensively, the competition for available computer time makes it seem doubtful that they would have been used at all had they been made any longer.

Some of the students were critical of the word contractions necessitated by the limited amount of computer memory. Such a memory restriction was not anticipated when the TPI programs were defined and eventually forced a reduction in the number of questions per program. A further reduction of questions per program would have eliminated the word contractions and actually doesn't seem to be such a bad alternative if textbook sections were used as an
integrating unit. The only disadvantage of reducing the number of questions per TPI program is the subsequent increase in the number of TPI programs to be handled.

The use of HP EDUCATIONAL BASIC as a TPI language was disappointing because its lack of character input capability and its restricted formatting of numbers caused some questions to have less than optimal readability. However, versions of BASIC are available that do solve these problems. In addition, the interactive nature of BASIC's INPUT statement and the imbedded calculation capabilities of the PRINT statement are quite useful in presenting mathematical questions. Furthermore, the use of the same language for TPI, quiz correction and program checking reduced the overall effort required in generating references to the student data banks.

With regards to the TPI activity itself, its value for this course with respect to improving achievement was minimal; however, the enthusiasm of the students and their helpful comments for improving the TPI methodology attest to its potential as an effective instructional and motivational activity.

Feasibility of requiring programs as the only form of homework. Too many homework programs were required for the course. Some students did enjoy the programs but the fast pace and time-consuming coding procedure eventually reduced the number of participating students to six. The value of the programs for teaching new concepts was not established but, at least for those students who did participate, it seemed that programming was more useful for refining or "polishing" concepts that had already been learned. The lack of participation in
programming by the majority of the students should not be considered a rejection of programming as a course activity but, rather, a rebellion against unreasonable expectations. Most of the students stated that they would have done more programming if they had had more time, been better prepared, and could have gotten on the computer more easily.

As a result of this study, one of the author's original justifications for the value of programming, that of forcing a complete understanding of the problem, seems to be suspect. There were some instances of a student missing a question on the final exam that was identical to a problem that he had programmed correctly. Occasionally, the author found himself, in the hustle and bustle of getting the check programs and sample programs written, scanning the text for a needed formula to solve a problem. Although feeling guilty about this lack of reading and understanding how the formula was derived, the expediency of getting the programs written overcame any academic considerations. It's quite conceivable that the students reacted in the same manner. Under these circumstances, the homework assignment becomes an exercise in programming skill and actually deters from the learning of the subject matter because of the non-productive time spent in programming.

In summary, the writing of programs as the only form of homework (other than reading the text) does not seem feasible. Additional activities such as theorem proving, solving word problems, and graph drawing are necessary and should be integrated with the computerized activities (Their use could still be monitored and prescribed by the student record programs). These additional homework activities are an
integral part of mathematics and should not be slighted because of the availability of a computer. In addition, their usage would allow alternatives to the non-computer-oriented student and should reduce the load on the computer.

Feasibility of check programs. The use of check programs for verifying program correctness was quite successful; however, the student's annoyance with the occasional errors in these programs emphasizes the need for their reliable and correct operation. The high ratio of programs completed to programs tried, and the covert efforts to obtain more than seven tries at a program, emphasizes the intensity of the student desire for generating a correct program.

The seven-try limit was originally instituted to discourage students from concentrating too much time on one problem and to encourage the students to do some hand checking before running their programs on the computer. However, since no data was collected on student programming procedures and, since the students did covertly try to override the limit, the value of such a limit is questionable.

Pencil-marked cards for programming. Although the H-P pencil-marked cards were specifically designed for BASIC programming statements, most students would have saved time either by using a keypunch or by being able to type their programs into the computer from a teletype terminal. This estimate of reduced time, however, assumes that the student would not have to wait in a queue at the keypunch or TTY. Therefore, although the generation of programs in computer readable form at home is an attractive solution to the problems of student accessibility and large
volume input, the cards designed by Hewlett-Packard seem to be a poor media for effecting this solution.

A few additional observations were made in the areas of quizzes, record keeping, testing, and student attitudes:

Quizzes. The quizzes were well received by the students and the students did quite well on them. They were a controlling force in keeping the course on schedule. In addition, the reporting of the number of students missing each question allowed poorly learned concepts to be identified and reviewed the next class session. These item analyses should also be helpful for eliminating poorly-worded and non-discriminatory questions in future classes. Even more useful item analyses could have been generated if the students' specific quiz responses could also have been saved.

Record keeping. Once running, the student record keeping was fairly reliable and accurate. One very important item of student data, the amount of time the students spent on homework, was not kept in the computer but was available from their quiz cards. If additional memory was available, this data should have been computerized and reported as a means of helping the students manage their time. It was an integral part of the results presented in Chapter Five and its recording should be seriously considered for any experimentation requiring out-of-class student activity. Although no checks were made concerning the accuracy of the time data, the correspondence of the programming activity and quiz results with time spent seemed to imply that the students were relatively conscientious in their recording.
Testing. Although multiple choice questions are convenient for computerizing the grading of quizzes and exams, this type of question is quite inadequate for measuring activities such as graph drawing, theorem proving, and algorithm generation. Therefore, if an experimental study is trying to compare algorithm generation to non-algorithm generation as a method of homework, it seems that the algorithm-oriented course is placed at a disadvantage if achievement is only measured by a multiple-choice item exam.

Attitudes. Despite the time constraints and developmental problems, half the students said they enjoyed the course and most said they would have enjoyed it with modifications. Those who were indifferent or disliked the course were either critical of the time constraints or of computers in general. The diversity of the attitudes expressed seems to indicate that an integrated computer course does offer an attractive alternative to conventional mathematics instruction for some students, but should not be required of others or, at least, only minimally. Possibly, with the help of their own comments, future implementations of the course could be designed to make computer usage more attractive to the anti-computer students.

Although the lack of academic pressure was appreciated by most of the students, it was abused by a few. Those that did abuse this privilege would probably not have done so if they had had adequate programming training, more available computer time, and alternative activities other than the computer-oriented. This freedom might not have been as well accepted in a less academically-oriented school than CCDS; however, even at CCDS, the students' occasional manipulation of their records
seems to indicate a misplaced emphasis on evaluation rather than learning.

In conclusion, although quantitative data to support or refute the value of this computerized course is not presently available, the diversity of activities offered and the availability of dynamic assessment and prescription warrants its further consideration as an instructional model. Having made the above statement, a certain obligation is assumed for responding to the question, "Where do we go from here?" There seem to be two alternative responses depending upon whether the reader a) accepts the model with reservations and would like to implement it, or b) thinks the model has possibilities but would like to have some comparative results before judging its value. The next section of this chapter (6.4 Recommendations for Future Classroom Implementation) is a response to the people in the first group; its following section (6.5 Suggestions for Future Comparative Evaluation) is a response to the people of the second group.

6.4 Recommendations for Future Classroom Implementation

The recommendations included in this section are intended to be general enough to apply to any mathematics course feasible for computer utilization; however, occasional references to analytic geometry in particular will be made. Although this study was conducted with a class size of only 18 students, the recommendations presented here are hopefully class-size independent. The following paragraphs will discuss recommendations for each of the components of the course; however, before these detailed recommendations are discussed, a few suggestions on organization considerations are presented.
Organizational considerations. The first suggestion for future implementations is that the time constraint of this study should not be adhered to, especially if the course is to be taught at the sophomore level. Obviously a school's scheduling procedures will somewhat influence the course length, as will the level of the students being taught, but probably the determining factors should be the course schedule recommended by the textbook author and the teacher's experience.

In organizing future implementations the tendency to use the computer "because it's there" should be resisted. Non-computer activities such as graphing, theorem-proving, etc. should not be slighted and ineffective computer activities should be discarded. This study only considered four of the seven activities described in Chapter One but some of the others could be integrated without much additional cost or development time. Particularly, in the area of demonstration of mathematical concepts, Hewlett-Packard offers relatively inexpensive (less than $4,000) graphic CRT's and hard-copy graphic plotters. These devices could be used for classroom demonstration, integrated with the TPI, or available for individual student investigation with preprogrammed routines.\(^1\)

In planning the integration of the computer activities, the following considerations should be paramount:

1) Coordination of all activities with a set of preestablished

\(^1\)An example of this last activity would be a program that would ask the student for numbers to be used as the focus and directrix of a parabola and then display or plot the parabola.
instructional objectives.

2) Estimating the effectiveness of each activity in achieving its desired objective.

3) Estimating out-of-class time required for each activity, both computer and non-computer.

4) Establishing a reasonable limit on the required amount of out-of-class time to be spent on the course.

5) Estimating the available computer capability for the course, including available terminal time.

6) Estimating the amount of time needed to generate the material to support the activities; this amount of time is a function of available software.

7) Estimating the amount of teacher time needed to support the activities of the course as it is in progress.

8) Estimating the amount of teacher time that will be available prior to and during the offering of the course.

9) Scheduling and integrating the activities within the constraints established in 2 thru 8. (This may require a computer!)

The establishment of instructional objectives for a complete course could be a momentous task for a teacher—if no outside assistance were available. Fortunately, curriculum committees, authors of textbooks, etc. explicitly or implicitly allude to such objectives when they are designing a course; and teachers from experience and/or departmental meetings and conversations with colleagues usually have rather firm commitments to satisfying a specific set of objectives.
Textbook sections were originally used as an organizing unit for this study, and although memory restrictions eventually forced the use of session number as an integrating unit, textbook sessions still seem the more preferable for integration of computer activities. Therefore, as the most pragmatic method of course organization by a classroom teacher, it is recommended a search be made for a textbook whose organization and contents are quite compatible with the course's instructional objectives; the satisfactory completion of specified sections of this text could then be used in place of the teacher's instructional objectives. The teacher would still have the option of selecting the appropriate sections and establishing the criteria for successful completion of these sections. Although more detailed objectives than textbook sections would be desirable, the cost of the additional computer storage required to maintain their interrelationships and related activities would probably not justify the added achievement attributed to their finer resolution.

After the appropriate textbook sections have been correlated with the instructional objectives, they should be combined and interrelated into a hierarchy of the type suggested by Gagne (1963).\textsuperscript{1} Such a hierarchy could then be stored with the prescription algorithm on paper tape or, preferably, in the protected memory area of the computer. The advantage of having this hierarchy permanently in the computer is that it could be referenced by the TPI programs and other programs that

\textsuperscript{1}Many textbooks already have such a hierarchy included in their preface or as part of the teacher's manual.
assess and prescribe student activity. The more detailed sections of
the recommended implementation will now be discussed. Although the
correlation of instructional objectives with textbook sections has been
recommended, the term "instructional objective" will be used for the
remainder of this chapter to keep the discussion as general as possible.

Computer to be used. Since the computer used for this study was a
minicomputer, the recommendations in this section will be only for a
minicomputer application. However, time sharing with its advantage of
1) reducing memory restrictions, 2) eliminating the need for assembly
language programming, and 3) eliminating the use of paper tape storage
for support programs\(^1\) is certainly an attractive alternative. The au-
thor considered calculating the cost of using time-sharing for the cur-
rent study but abandoned the effort because of non-comparable units and
the multitude of pricing schemes. From discussing costs with time-
sharing users and vendors, a reasonable comparison would place time-
sharing costs within $100 per month of the minicomputer costs.

However, the cost of minicomputers is steadily decreasing\(^2\) and
their advantages described in the previous section make the decision of
minicomputer versus time-sharing a very difficult one. An equally dif-
ficult decision, assuming only a limited amount of available

\(^1\)Not only could the use of paper tape storage for these programs
be eliminated, the total memory required for the programs could be re-
duced by only storing one copy of statements common to all sets of pro-
grams; i.e., the START statements for check programs and TPI.

\(^2\)The HP2114B with 8K memory used during the study can now be re-
placed by a new H-P minicomputer with twice the memory and no increase
in cost.
funds, would be whether to add a high speed paper tape punch, a graphic CRT, a graphic plotter, or additional memory. Such decisions can only be made by determining the costs of all alternatives and coordinating the requirements of the computerized mathematics courses with the needs of the other computer users of school.

Text. The textbook chosen for this course was well-liked by the students, teacher, and author and is recommended for future analytic geometry implementations. However, this recommendation tacitly assumes that future implementers of the course have instructional objectives similar to the implicit objectives of the text.

Orientation. An orientation lecture (or lectures) is a must for a course of this type. Not only should specific computerized activities be discussed, but the entire philosophy of the course should be presented to the students, including a list of the desired instructional objectives. In addition, having the students interact with sample TPI and investigative programs may reduce their reluctance to use them as the course progresses. Probably at least two class sessions should be allocated for orientation; the second session should include a short quiz on the course organization followed by a question-and-answer period.

Student Records. The arrangement of student data in the protected region of memory was quite adequate for the course given but would have to be slightly redesigned if the course is to be organized in terms of instructional objectives. Particularly, since each quiz or TPI question would be related to a specific objective, the current reduction of the quiz and TPI data to the "total number missed" is inadequate
and should be replaced by the student's score on each item.

The reporting programs could then be used to sum the item results into the student's score for the activity. Actually this redesign would not significantly affect the amount of memory needed because the number of bits per quiz per student would only have to be increased from 4 to 8 (1 for each of 7 quiz items and 1 for attendance) and would not have to be increased at all for the TPI (number of tries and number wrong would be replaced by the student's score on each item).

A much larger drain on protected memory would be the storing of the instructional objective hierarchy. This information could be saved on paper tape as part of the reporting programs but at a loss of prescription immediacy. Additional memory would also be required if more students and activities were included. One method of increasing available memory is to reduce the student's programming record from 4 bits to 1 bit per program since the saving of the number of tries of a program is not necessary for the prescriptive process, although it is useful for analyzing program difficulty and student programming competence.

Assuming more memory were available, the saving of the student's exact choice for each quiz item would allow the use of a more sophisticated prescription algorithm. The prescription algorithm and teachers report could also be improved if the student data included a record of activities prescribed and the student's success in applying the prescription. If, in addition, pretest results and homework times could be saved in the data bank, more information would be available for determining the proper prescribed activities.
Prescription Process. The prescription program, or subprogram, should access the instructional hierarchy; this hierarchy will include interrelationships of instructional objectives and tables linking these objectives to quiz items, TPI questions, reading assignments, text questions, homework programs, investigative programs, and any other activities. Since the instructional objectives will range in difficulty and will be interrelated, the prescription algorithm may require the student's results on interrelated or supportive objectives before selecting an activity, or activities, to assist the student in attaining a specific objective. Hopefully, the same prescriptive algorithm could be used for all objectives with the input to the algorithm being each specific objective's associated activities and related objectives.

An example of an evaluation of a student's quiz results might help illustrate the recommended prescriptive process. Suppose the student missed questions 1, 4, and 6 on a daily quiz. The incorrect response to question 1 may identify the lack of attainment of a relatively straightforward objective; the subsequent prescription might be to reread pages xxx thru xxx of the text and then take TPI number xxx. Question 4, however, may encompass three or four objectives and therefore an incorrect response may necessitate a search of the supporting objectives to ascertain the student's weakness. This search may reveal that the student's weakness was the objective assessed in question 1 in which case no further action would be necessary. However, the student's weakness may have been with an objective he had already attained; in this case a simple review of the textual material would probably be sufficient. Another possible result of the search
would be that the student had attained all supporting objectives and was just having trouble using them in an problem-solving situation. The prescription for this type of weakness might be to either prove a theorem, draw a diagram, interact with an investigative program, or, if an algorithmic approach is warranted, suggest that the student program a problem. The missing of question 6 would require a similar analysis.

The suggested remedial prescriptions would, in most cases, be supplemental to the daily homework assignments. The daily assignments would be those activities that have demonstrated the most overall success in attaining the desired objectives for the coming class session. Where the difference in effectiveness of two or more activities is relatively inconsequential, the students should be given their choice. If a student did poorly on a quiz and the required homework, his remedial activities may be so time-consuming that it would be unreasonable to expect him to also do the required homework. In these cases, a less time-consuming required homework for the next session should be prescribed.

The above example of the prescriptive process implied a dichotomous assessment of the attainment of an objective. For a first attempt, this should be more than adequate; however, with developmental experience and increased memory capacity, a more descriptive assessment of the attainment of an objective might be used. This could be an "attainment index" which would be increased by successful activities and decreased by questions missed, excessive programming tries, etc. The prescriptive process could then be modified to prescribe
different activities depending upon the value of the "attainment in-
dex."

Reports. One of the problems with the course taught at CCDS was that
the pressure for available computer time kept the students from using
the "Student Report Program." This lack of feedback to the student was
eventually remedied by Mr. Klitz posting the "Teacher Report" on the
bulletin board. This remedy, however, generated its own problems such
as lack of anonymity, jealousy, and pressure to have a good record.
The public posting of student records is a teacher's perogative, and in
this study an imperative, but if other means are available for present-
ing the student record to him privately, they should be used.

An alternative to solving the individual reporting problem is to
modify the teacher's reporting program to list each student's record
and prescription separately and to allow enough blank lines between
the reports for the teacher to separate them for daily distribution to
the students. These reports would probably print faster than the
current teacher reports because they would only report data on activ-
ities tried. The report for the teacher could still be the current
teacher report, except organized with respect to instructional objec-
tives. However, if time were at a premium, the teacher should be given
a choice of asking for shorter management-by-exception or summary re-
ports. Of course, nothing should prevent the teacher from asking for
all three reports except, possibly, the denial of computer time to the
students while the reports are being printed.

1The computer could even generate a cutting line by printing a
row of hyphens.
The student reports and prescriptions generated in the above manner would be very useful but would only remain valid until a student attempted his required activities. This requirement for currency was the reason that the Student Record Program was incorporated into this study. However, since the results of this study suggest that a separate program would not be used, one alternative is to have the prescription and reporting programs stored permanently in the computer. This, however, would require additional computer memory. A less memory consumptive, but more difficult to implement, alternative is to incorporate prescriptive and reporting algorithms in the homework checking, TPI, and investigative programs.

Quizzes. Although there were some initial difficulties in using the prepunched Hewlett-Packard cards for recording quiz answers, these cards still offer the most convenient medium for acquiring classroom generated data in computer-readable form. Although some time (both classroom and out-of-classroom) was wasted at the start of the course due to poorly marked cards, this time was not really completely wasted because the students were learning how to interface with a computer information system. However, the slope of the learning curve for some students was much smaller than should be tolerated.

For future implementations, the possibility of purchasing cards designed especially for quizzes might be considered. Usually specially printed cards, if ordered in large enough quantities, are not very much more expensive than standard ones. Likewise, some of the teacher time consumed in editing the cards and running the quiz correction program could be eliminated by hiring student operators or
assigning the duty to a student who marked his card incorrectly.

Although multiple-choice items were used exclusively for the quizzes in this study, other quiz items such as definitions, completions, graphs, etc. may be used. These questions would be discussed in class after the quiz and then recorded as "1" or "0" on the card depending on the code for "correct" or "incorrect". The results of these questions would then be placed in the student record and used in the same manner as the results of the multiple-choice items. Most teachers are experienced enough to develop procedures to guarantee that this data would be recorded honestly.

TPI. A well-written textbook is a very effective method of presenting new information to a majority of students. When preceded or followed by an illuminating lecture, it offers an instruction model of common acceptance and proven success. Thus, whenever possible for future implementations, this combination should be used for the initial presentation of material. However, since not all textbooks are well-written (or read), and not all lectures are illuminating (or heard), some alternative is necessary for students who are not attaining the desired objectives. For attaining objectives that are not of a problem-solving nature, one recommended alternative is the use of short remedial TPI programs such as those used in this study.

The major difficulty with the TPI programs developed for this study was that 1) they really didn't present very much new information because of text limitations and 2) they didn't remedy any specific weaknesses because they were not used when recommended. The prescriptive algorithm did recommend a related TPI program if a
student obtained a poor quiz score or was working ahead, but the unit
used for coordinating these activities was a session number rather
than a instructional objective. If future TPI usage is to be organized
with respect to textbook sections (which is strongly recommended), many
more TPI programs will be needed and more complex logic must be used
to present the questions in a hierarchical order. As an example: If
a student continually misses a particular TPI question, even after
being presented remedial messages, it would be a waste of his time,
and the computer's time, to present the remaining TPI questions
which assume a correct answer to the troublesome question. In this
situation, the TPI program should recommend that he take (or retake)
an earlier TPI program, reread the text, possibly confer with his
instructor, and then retake this same TPI program.

An additional degree of complexity would be added to the TPI pro-
grams by analyzing a student's incorrect answers and then presenting
him questions related to the poorly learned objectives as identified
by the incorrect response. Unfortunately, this type of analysis would
require more questions and more programming logic than could have
been supported by the 8K memory of the HP 21L4B. An additional
8K of memory could tremendously increase the effectiveness of the
TPI programs, not only by improved analysis and text messages, but by
reducing the number of paper tapes to be handled by the teacher and
students.

**Homework programs.** Programs should only be required as homework for
those instructional objectives that require the synthesis of less
difficult objectives or are quite algorithmic in themselves. The majority of the homework programs specified in this study would fit into one of these two categories but too many were required for the length of the course. The required number of programs for future implementations should be consistent with the desired average homework time and with the estimated programming time for the problems. The student's comments on programming seemed to indicate a willingness to do up to three programs a week.

The assumption that students would use previous programs as subprograms for later programs was not evaluated. Although this assumption may still be valid, some of the students expressed a desire for a set of available library subprograms. These subprograms would, in most cases, be correct solutions to earlier problems but, occasionally, would include routines difficult or too mundane to be required of the students. They would be incorporated into the check programs and instructions for their use would be documented in the problem statements. The use of these subprograms would also result in the "computation mode," defined in Chapter 1, becoming part of the integrated course.

It was also assumed that the students would have adequate programming skills to solve the problems without being severely hampered by syntax and logic errors; this was not the case. Therefore, it is recommended for future implementations that programming skills be assessed before the course begins, and sufficient activities, possibly programmed instructions manuals or TPI, be made available to the students so that they may quickly obtain at least a minimal level of
Rather than restricting student programming attempts to seven per problem, a better approach might be to assess their attainment of the objectives assumed to be necessary for solving the problem and deny them usage until the prerequisite objectives are attained. This denial of access to the check programs, however, may be rather demoralizing. An alternative could be to allow the students one initial try and, if not successful, then prescribe remedial activities to assist their attainment of the prerequisite objectives.

Additional activities. Although programming homework problems was justified in the preceding paragraphs as a means of attaining the more complex instructional objectives, a possible alternative for attaining these objectives is the use of investigative programs. These programs would be similar to the required student programs but would be pre-written and would only require input from the students. They could be used over and over for "what if" analyzes and could incorporate a CRT or graphic plotter for output. The use of these programs may not be as effective as the programming problems, but they do offer an alternative to the anti-programming student. In addition, they are a less time-consuming activity than programming and could be used during those weeks in which the maximum number of programs has already been assigned. If investigative programs are adopted for future implementations, however, a few trial runs of some sample programs should be given before the course begins to familiarize the students with their use.

An even less-time consuming activity for complicated objectives would be to assign written homework problems of the type that would
require the student to supply answers to all possible alternatives. As an example, homework problem 10 of this study required the student to write a program to identify the type of quadrilateral formed by four sets of input coordinates. There are only seven possible categories of output. A written homework assignment to replace this program would be a seven-part question with each part containing a set of coordinates that represent one of the seven categories. The student would still be asked to identify the type of quadrilateral. These assignments may not be as effective as programming problems or using investigative programs, but they could be used for variety or, as a last resort, if time becomes severely limited.

**Operational procedures.** The use of student operators for future implementations would be desirable but not completely necessary. If a choice had to be made between having a student operator or a quiz editor, the quiz editor should be selected. However, many schools do have business operations courses and computer clubs and, therefore, the obtaining of a student operator might not be difficult. With regards to maintaining operational capability, the assistance of a student computer expert, not necessarily an operator, is especially desirable.

Regardless of the availability of a student operator, a schedule of available computer time should be posted in the computer room and monitored by the operator or the classroom teacher. One possible scheme for allocating available time, used by many golf courses, is the separation of each 15-minute period into 2 five-minute periods for scheduled one-student use, and 1 five-minute period on a first-
come, first-served basis. The dedicated five-minute periods could be
used for TPI or investigative programs and the free five-minute periods
could be used by many students for program debugging. If a student
wanted more than one five-minute period using this scheme, he could
sign up for a second period; however, this time could be overwritten by
a student who hadn't signed up yet and wanted that time.

Testing. Ideally, if the quiz items and other computer activities ac-
curately assessed the attainment of the instructional objectives of the
course, no final exam would be needed. Certainly final exams, with
their inherent time constraints, can not possibly assess the attainment
of all the course objectives; consequently they usually only concen-
trate on those objectives that can be easily and quickly measured. How-
ever, final exams do provide excellent motivation and offer an addition-
al assessment. In addition, their elimination might result in an over-
emphasis on getting TPI questions and homework assignments correct and,
consequently encourage manipulation of records and borrowing of home-
work. Therefore, it is recommended that a final exam be given but its
results used in a rather novel way, namely, correlating exam scores
with attainment measures in the student records to generate a student's
"credibility index." The results of the final exam would hold for the
objectives measured and the student's record, modified by the credibi-


167

lity index, would hold for the objectives not measured. The student's
overall grade, assuming one is necessary, could then be determined by
relating the student's success on each objective to some weighted mea-
sure of the value of that objective. This concludes the recommenda-
tions for future classroom implementations.
6.5 Suggestions for Future Comparative Evaluations

If a comparative evaluation is desired, the experimental treatment should include the recommendations of the previous section. The control treatment could be the conventional text, lecture, written-home-work approach oriented toward attaining the same instructional objectives as the computerized course. An alternate control treatment could be a course using only one of the computer activities. Ideally, a multi-treatment experiment comparing all possible combinations of computer activities to the integrated and non-computer approach would be the most valuable. However, it is unlikely that enough students and facilities could be obtained to attempt such an experiment. Regardless of the design of the experiment, students should be randomly assigned to each treatment.

The following paragraphs include some suggestions for data collection, testing, and operational procedures. Although these suggestions are primarily oriented toward a comparative evaluation, they should also be helpful for a non-comparative replication of the study to further improve the design and implementation of the model.

Development of course materials. Although the supportive programs used in this study are available for future courses, these programs would have to be modified and many more programs would need to be written, especially if the course is lengthened and investigative programs and additional TPI are included. If the experimenter has access to a large computer, one time-reducing alternative for program development is to use the large computer to generate the paper tapes
to be used on the minicomputer.

From the development times reported in Chapter Five and Appendixes B and C, and considering his own personal schedule, the future experimenter should be able to estimate when he must start programming the support programs to guarantee their error-free operation when the experiment begins. However, since time for development fluctuates from person to person and, since little information is available for estimating this parameter, each future experimenter is encouraged to keep a time history of each significant development event. A significant event might be the start-stop dates (times) of writing the homework assignments, a particular check program, or a TPI. With regards to the supportive programs, it may even be useful to include the start-stop times of events such as initial writing of the program, typing the program into the computer, debugging, and generating paper tapes and program listings.

Preassessment. In order for the results of future experiments to be more precise, the programming skills of the subjects should be measured before they begin the course. In fact, since many of the homework programs and homework time limits are based on an implied level of programming expertise, a minimum programming competence should be required. If a student has not reached this minimum level, he should be given remedial assistance and a time limit in which to learn enough to pass the minimum level on a comparable retest. This approach has merit, not only for enhancing the probability of success of the computerized course, but it also for eliminating the frustration of those students who do not have the required skills.
In addition to testing for programming skills, a standardized test measuring programming aptitude should be administered to the students of both the control and experimental groups. This data should help to assess whether aptitude measurements should be used for determining future assignments of students to computerized or non-computerized courses. If a correlation was established between programming aptitude and course achievement, future implementations might even want to consider programming aptitude as part of the prescription algorithm.

With respect to the course material itself, achievement can not really be measured without some assessment of the students' knowledge of the course material before the course begins; unless, of course, it can be safely assumed that the students had no previous knowledge. Since it has been recommended that future courses be organized with respect to instructional objectives, student achievement should be measured by identifying those objectives attained at course completion which were not attained at course initiation. Therefore, a preassessment of the student's attainment of the course's instructional objectives is necessary. Because of the anticipated large number of these objectives, the preassessment must be carefully constructed to test as many of the objectives as possible in the allotted time. A computerized branching test as recommended by Ferguson (1970) would be one possible approach to satisfying this requirement. Even if it could not be computerized, a pencil-and-paper branched test still seems to offer excellent potential for assessing the most objectives in a given period of time.
Data Collection. To properly assess the comparative effectiveness of a computerized versus a non-computerized version of a course, the amount of time expended by the students of each approach is a very significant parameter. During the orientation lecture(s), the students of both classes should be convinced of the importance of accurate reporting of this data. In fact, it would be advantageous for the students of both groups to report the time expended for each activity. During the execution of the study, this data should be daily monitored for encoding errors or lack of accuracy, i.e., a student reporting the same amount of homework time each day.

Because of the computerized course's own data requirements, it can be anticipated that very little protected storage will be available for purely experimental data. One alternative is to write a program to collect and store experimental data and to generate a daily experimenter report. This program and its data would be read in via the high-speed paper tape reader. After printing the report, the revised data would be punched out on the TTY punch unless a high-speed punch was available. Since this program would be run only once a day, and hopefully not during school hours, the storing of data on paper tape seems feasible.

The suggested daily experimenter report would include time data, activities completed that day, and analyzes of the students' adherence to their prescribed activities. It could also list those activities with no student utilization or, if enough data were available, identify those activities that were requiring an inordinate amount of student time. Whether in-course adjustments should be made as a
result of these analyzes would be the experimenter's prerogative.

Operational Procedures. The signup sheets recommended for future implementations should also be used for any future experimentation. Besides their scheduling advantages, these sheets are an excellent source of data and, particularly for the experimental version, they should request the student to identify the activities tried and to comment on any difficulties encountered.

For proper reporting of the developmental and operational problems, it is recommended that an operator's log be kept. This log should include time and duration of machine malfunctions or support program errors. In addition, it should include space for comments on scheduling and operational procedures. This log should be maintained by the lead student operator or, if no student operators are available, jointly by the teacher, experimenter, and student computer expert.

Posttest. Rather than compare results to a standardized test, it seems more consistent with the philosophy of the course to use the same examination procedure as that recommended for future implementations. However, if comparison to standardized achievement tests is desired, one could compare results on common instructional objectives. Identical final exams should be given to both the experimental and control group and used in the same manner (i.e., adjusting attainment level by credibility index) to determine overall course accomplishments. Since the control group's records are not computer-stored, the final reporting of objectives attained may be a rather laborious task, depending, of course, on the number of measurements taken.
Although the relative attainment of instructional objectives between the two approaches (or absolute attainment of objectives if no control group is available) is the criteria for evaluating the effectiveness of the computer approach, a residual payoff of this approach should be an improvement in programming skills. Therefore, the programming pretest, or an equivalent form, should also be administered following the posttest.

Student evaluation of course. In addition to the items used on the questionnaire for this study, the post-course student questionnaire should include comments on 1) the pretest, 2) computer availability, 3) signup procedures, 4) value of student and teacher reports, and 5) suitability of prescriptions. A similar questionnaire, excluding, of course, comments on the computer activities, should be given to the control group. Since students do compare approaches during the course of the experiment, it might be interesting to ask if the students of either group would have preferred to have been placed in the other group.

If the experiment includes a large number of students, the use of mark-sense cards might be considered as input to a special program written to process and report the student's evaluations. Hopefully, the computerization of this data would not deemphasize the value of written student comments which should still be collected.

Analysis of Results. If instructional objectives are weighted and attainment levels are quantitized, it should be possible to statistically compare the results of the experimental and control group.
6.6 Recommendations for Future Research and Development

This chapter will conclude with a tabular presentation of recommendations for classroom teachers, doctoral degree candidates, research centers and professional societies, and computer companies.

Classroom Teachers. It is recommended that teachers:

1. report successful and unsuccessful procedures for allocating available computer time.

2. report successful and unsuccessful prescriptive algorithms regardless of whether the data or activities are computerized.

3. report successful and unsuccessful computerized activities oriented toward specific instructional objectives.

4. report for each TPI program
   a. language used
   b. branching scheme
   c. number of statements
   d. time to develop
   e. average student contact time
   f. judgement of its effectiveness.

5. report for each homework program assigned
   a. language used
   b. type of computer input
   c. number of statements
   d. time to develop check program, if any
   e. average number of student tries before getting correct
   f. average time spent on program
   g. judgement of its effectiveness.

6. Compare effectiveness of programming problems that just require formula evaluation to doing pen-and-pencil exercises using the same formula.

7. Compare results of a course with programming requirements evenly distributed, say two programs per week, to a course with fewer programs at the start and more near the end.
8. Compare programming results for students who have access to preprogrammed course-related subroutines to results for students who have to generate any needed subroutines themselves.

9. Report lack of information and background weaknesses that inhibit the teacher's effective development and execution of a computerized course.

10. Report any efforts toward enlisting the assistance of members of a computer science course or club in course development.


The recommended reporting could be presented at professional or computer user-group conferences at the local, state, regional or national level, or through journals associated with these organizations.

Doctoral candidates. The following questions have evolved from the author's participation in the development of the course material and the evaluation of the results. Although the study had many limitations, which inhibit any attempts at generalizing the results, the author would have to answer "yes" to these questions if they were asked of him at this point in time. However, many people would disagree with this answer, or respond that there is not enough information available, and therefore these questions are presented as a means of encouraging research to make available more information.

Q1: Will a computer-oriented course show more achievement and improved attitudes if scheduling of computer usage is required?

Q2: Will a course with a text for initial presentation of material and TPI for review require less total computer time and less student participation time than a course with TPI alone, while not decreasing achievement?

Q3: Will an integrated computer course with a selection of available instruction alternatives be more effective if a prescriptive process is included?
Q₄: Will a computerized course have significantly greater dispersion of time spent by students than a non-computerized course?

Q₅: Will the availability of daily individual student records increase the overall achievement of the class?

Q₆: Will the requirement of homework problems to be programmed on a computer not significantly increase the overall achievement of the class on any standardized tests?

Q₇: Will a course with short daily quizzes result in a higher mean score on a final exam than a course without quizzes. This prediction assumes the quiz items and exam questions are similar to content and required response?

In addition to testing the questions presented above, the following investigations are recommended for consideration by doctoral candidates:

1) Perform comparative evaluation recommended in previous section.

2) Replicate study for other math courses.

3) Replicate study eliminating different activities.

4) Replicate study adding different activities and/or computer components, i.e., additional memory, high speed paper tape punch, graphic CRT, plotter, etc.

5) Replicate study using time sharing while maintaining all other components.

6) Replicate study comparing the results of lecture versus no lecture. No lecture would have same class time but only for answering questions; course material would be presented via text or programmed instruction.

7) Generate and compare the effectiveness of different prescriptive algorithms while maintaining the same activities. Of particular interest are the questions:

   a) should a student be assigned a problem to program if he hasn't attained the required objectives?
b) should a student be assigned a programming activity to help attain complicated objectives and to enhance his knowledge of the related elementary objectives, or should he be required to concentrate on other elementary objectives that he is having difficulty attaining?

c) what importance, if any should the prescriptive algorithm assign to programming aptitude?

8) Compare results for remedial TPI with text first to remedial TPI with text presented only if a question is missed. Include comparisons of terminal contact time.

9) Compare results of requiring TPI for all students instead of only for those prescribed.

10) Compare effectiveness of programming homework problems to interacting with an investigative program.

Research centers and professional societies. Considering the myriad of activities, such as audiovisual aids, math labs, programmed instruction, computer-oriented, etc. now available for mathematics instruction, it is becoming exceedingly difficult for a teacher to learn "how" to use these activities, let alone determine "when" to use them. It is inconceivable that any one of these activities is best for all objectives for all students and therefore a teacher is faced with the momentous task of determining which activity to use when. Compounding this problem is a varying degree of specificity among organizations in identifying and interrelating instructional objectives. The resolution of this problem is not one to be addressed by a single researcher but requires the expertise and coordination of the organizations responsible for influencing the progressive development of mathematics instruction. Therefore it is recommended that these organizations:
1) address the problem of defining what is an "instructional objective."

2) determine what is a manageable number of objectives for a course in general and how intricate a hierarchy of objectives is feasible.

3) coordinate efforts to identify and accept the objectives implicit in the courses defined by these organizations.

4) support research in developing valid and reliable testing procedures for determining if the accredited objectives have been attained.

5) support research in identifying which activities are most effective for a given category of student pursuing each accredited objective.

6) initiate developmental efforts in those activities that have been very effective for some objectives but not even tried with others.

7) encourage textbook publishers to identify the objectives of their texts and to include in their teacher's editions, the interrelationship of these objectives and recommended activities for attaining them.

8) establish a computer-maintained information system containing a) the descriptions of accredited objectives, b) their interrelationships, c) a list of effective activities in attaining these objectives, and d) a list of questions that validly measure if the objective has been attained. This information system should also include a cross-referencing of objectives to textbooks, standardized texts, recommended courses, and research efforts.¹

With regards to efforts more germane to a computerized course, the research centers and professional societies should

9) support research in determining the proper age, or some other criteria, for requiring and teaching programming and computer utilization.

¹The Department of Defense (DOD) in the early sixties recognized the need for such a system to identify and coordinate its research and development (R&D) activities. This system is called the DOD 1498 File and includes definitions of each approved (on a 1498 form) DOD R&D activity and their interrelationships. These 1498's are used in almost all DOD information systems.
10) cooperate with the professional computer organizations in establishing standardized tests for measuring programming ability in the languages most commonly used for programming in the secondary schools, i.e., Fortran and BASIC.

11) cooperate with computer users groups, manufacturers, and software firms to a) help identify major technological constraints inhibiting educational applications and b) influence future hardware and software developmental efforts to eliminate these constraints.

12) survey the computer systems available to schools and report costs and advantages of these systems in the professional journal. This survey would have to be performed periodically to keep the information current.

13) offer professional seminars in specifying and selecting the proper computer resources for a given school environment.

Computer manufacturers and software firms. Most computer manufacturers are convinced of the tremendous potential of the computer as an instructional resource. However, they have decided, as have many teachers and school boards, to limit their commitment of resources until they are better able to identify those applications that have the greatest probability of acceptance by the educational community. Assuming that the activities used in this study will be accepted by the mathematics education community, it is suggested that the computer manufacturers consider the following recommendations:

1) design their operating systems, or monitoring portions of their language interpreters, to allow protected regions of memory for storing student records. The use of these regions should not require assembly language programming.

2) offer generalized student data base management systems. These systems should only require the teacher to identify the characteristics of the parameters to be saved and the equations of the prescriptive algorithms. Such a system could also allow limited queries and generate standard reports. This type of generalized data management systems is widely used in industry and government for the same reasons as suggested here; quick implementation and minimum required programming.
3) design both media (cards, forms, etc.) and devices (card readers, OCR, etc.) for allowing immediate and error-free input of quiz and test results to the computer.

4) develop libraries of check programs for courses requiring algorithmic problem-solving. These libraries could be included in the hardware cost or purchased for a nominal fee. They could also be developed by special interest committees of the computer user-groups.

5) in addition to check program libraries, the development of the following course-oriented libraries should also be considered:
   a) short TPI programs
   b) investigative programs
   c) subprograms for solving routine mathematic operations.

The above suggestions are not restricted to minicomputer manufacturers, but are directed to any computer manufacturer who plans to market computer systems designed for educational use.

This concludes the written portion of the report. Although not all the original objectives were obtained, a qualitative judgement of the results would suggest that an integrated, computerized course has potential for improving mathematics instruction and should be further investigated. Hopefully, this report will facilitate these future investigations.
BIBLIOGRAPHY

This bibliography will be separated into two sections: references cited in the text and referenced uncited but used in forming summarizations. The Publication Manual of the American Psychological Association, 1967 Revision, was used as a guide in formatting the references. This manual was carefully adhered to except for the following modifications that were made to assist the reader in obtaining the reference:

1. References to journal articles will include month and year as well as volume and issue number.

2. References to Dissertation Abstracts International will be of the form "Feb. 1971, 31A-3997." where 31A is the volume number and 3997 is the page number.

3. References to items in the ERIC (Educational Resources Information Center) library will be followed by the access number, e.g., E047962.

4. References to brochures from manufacturers will include publication numbers if available.

5. The abbreviation NCTM will be used for The National Council of Teachers of Mathematics.

Because of the length of the bibliography, the references will not be numbered consecutively; rather, they will be labeled by a designator of the form "A7" where A7 is the seventh item whose author's name begins with A.
CITED REFERENCES


B16. Bunderson, C. V. Comments during panel discussion on Computer science contributions to learning and teaching at the meeting of the Association of Computing Machinery, Chicago, 1971.


C10. CRICISAM (Center for Research in College Instruction of Science and Mathematics). *Calculus: A computer oriented presentation*. Tallahassee, Fla.: Florida State University, 1969. (Five volumes)


H1. Hahn, S. W. discussing experiences with CRICISAM calculus at Wittenberg University during Ohio sectional meeting of the Mathematical Association of America, Bowling Green, Ohio, May 1970.


H11. Hicks, B. L. CIDAC - a concept in educational technology. Audiovisual Instruction, Sept. 1971, 16(7), 68-70.


J3. Johnson, T. Learning how to apply the first true random-access learning system. Audiovisual Instruction, Oct. 1971, 16(8), 78-81.


Filmed as received
without page(s) 189

UNIVERSITY MICROFILMS.


UNCITED REFERENCES


D2. Dean, P. M. Learner versus teacher controlled arithmetic practice. San Jose, Calif.: IBM Corporation. ED 047956


H2. Hagerty, W. K. D. Development and implementation of a computer managed instructional system in graduate training. Dissertation
193

Abstracts International, March 1971, 31B-5308. (Florida State University, 1970)


M5. Molenda, M. H. How a school system gets into CAI. Audiovisual Instruction, June 1971, 16(6), 86-87.


# APPENDIXES

## APPENDIX A. QUIZZES

<table>
<thead>
<tr>
<th>Contents</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.1 Daily quiz sheets</td>
<td>199</td>
</tr>
<tr>
<td>A.2 Quiz item analysis</td>
<td>217</td>
</tr>
</tbody>
</table>

## APPENDIX B. HOMEWORK

<table>
<thead>
<tr>
<th>Contents</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>B.1 Homework assignments</td>
<td>218</td>
</tr>
<tr>
<td>B.2 Table of check program characteristics</td>
<td>228</td>
</tr>
<tr>
<td>B.3 Table of check program usage</td>
<td>229</td>
</tr>
<tr>
<td>B.4 Sample check program</td>
<td>232</td>
</tr>
<tr>
<td>B.5 Sample correct solutions</td>
<td>233</td>
</tr>
</tbody>
</table>

## APPENDIX C. TERMINAL PRESENTED INSTRUCTION

<table>
<thead>
<tr>
<th>Contents</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>C.1 Table of TPI usage</td>
<td>234</td>
</tr>
<tr>
<td>C.2 Sample printout of TPI No. 8</td>
<td>235</td>
</tr>
</tbody>
</table>

## APPENDIX D. STUDENT RECORDS

<table>
<thead>
<tr>
<th>Contents</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>D.1 Letter to Jeff Spain</td>
<td>236</td>
</tr>
<tr>
<td>D.2 Flowcharts of reporting programs</td>
<td>245</td>
</tr>
</tbody>
</table>

## APPENDIX E. TESTING

<table>
<thead>
<tr>
<th>Contents</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>E.1 Pretest</td>
<td>252</td>
</tr>
<tr>
<td>E.2 Pretest summary</td>
<td>255</td>
</tr>
<tr>
<td>E.3 Final exam</td>
<td>256</td>
</tr>
<tr>
<td>E.4 Final exam summary</td>
<td>263</td>
</tr>
</tbody>
</table>

## APPENDIX F. EVALUATION

<table>
<thead>
<tr>
<th>Contents</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>F.1 Evaluation sheet</td>
<td>264</td>
</tr>
<tr>
<td>F.2 Evaluation summary</td>
<td>265</td>
</tr>
<tr>
<td>F.3 Student comments</td>
<td>267</td>
</tr>
<tr>
<td>F.4 Teacher evaluation sheet</td>
<td>278</td>
</tr>
</tbody>
</table>
APPENDIX G. PROCEDURES

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>G.1 Quiz instructions</td>
<td>279</td>
</tr>
<tr>
<td>G.2 Quiz deck setup</td>
<td>280</td>
</tr>
<tr>
<td>G.3 Homework instructions</td>
<td>281</td>
</tr>
<tr>
<td>G.4 TPI instructions</td>
<td>282</td>
</tr>
<tr>
<td>G.5 Operating instructions</td>
<td>283</td>
</tr>
<tr>
<td>G.6 Sign-up sheet</td>
<td>284</td>
</tr>
</tbody>
</table>
A.1 Daily quiz sheets

Computerized Analytic Geometry

Quiz #1

A. Given $A = \{a,b,c,d\}$ and $B = \{c,d\}$

Then
1. $A = B$  1. $A \cap B$ is disjoint
2. $A \cup B$  5. $A = \emptyset$
3. $A \subseteq B$

B. $A = \{a,b\}$ has how many proper subsets?
1. 0
2. 1
3. 2

C. Which one of the following statements are true?
1. $A \subseteq B \Rightarrow A \subseteq B$
2. $(A \cap B) \subseteq A$
3. $A \cap B \Rightarrow A \cap B = \emptyset$
4. $A \cap A \cup B$
5. $A \cap B$ and $B \cap A \Rightarrow A = B$

D. The negation of "all birds fly" is
1. Only Dodo birds cannot fly
2. No birds fly
3. If you are a bird, you can fly
4. There is at least one bird that cannot fly
5. If you are a bird, you cannot fly

E. If the universe is the set of integers, then
\[ \{x \mid x^2 < 5 \text{ and } x \leq 0\} = \]
1. All negative integers
2. $\{0, -1\}$
3. $\{0, 1\}$
4. $\{-1, -2\}$
5. $\{0, -1, -2\}$

F. Given $P_1(-3)$ and $P_2(-6)$, the distance between $P_1$ and $P_2$ is ___, and the directed distance from $P_1$ to $P_2$ is ___.
1. 3, 9
2. 3, -9
3. 3, 3

G. 

The graph of which of the following sets (Assume the universe is the set of integers)
1. $\{0, -3, 1\}$
2. $\{x \mid x > -3\}$
3. $\{x \mid x^2 - 2x + 3 = 0\}$
4. $\{x \mid x = -3 \text{ and } x = 1\}$
5. $\{x \mid x^2 + 2x - 3 = 0\}$
A. The contrapositive of "If \(a \cdot b = 0\) then \(a = 0\) or \(b = 0\)" is
1. If \(a \neq 0\) and \(b \neq 0\), then \(a \cdot b = 0\)
2. If \(a = 0\) or \(b = 0\), then \(a \cdot b = 0\)
3. If \(a \neq 0\) and \(b \neq 0\), then \(a \cdot b \neq 0\)
4. If \(a = 0\) and \(b = 0\), then \(a \cdot b = 0\)
5. If \(a \neq 0\), then \(a \cdot b \neq 0\)
6. If \(a = 0\), then \(a \cdot b = 0\)

B. The set \(\{x \mid a < x \leq b\}\) is called
1. The open interval from \(a\) to \(b\)
2. The left open interval from \(a\) to \(b\)
3. The right closed interval from \(a\) to \(b\)
4. The directed distance from \(a\) to \(b\)
5. The closed segment \((a, b)\)

C. The graph of

\[
\begin{align*}
&\begin{array}{c}
\text{is the graph of}
\end{array}\\
&\text{is the graph of}
\end{align*}
\]

D. Which of the following describes the set of points which lie to the right of \(P_1(-4)\) together with the points which lie to the left of \(P_2(6)\), including \(P_1(-4)\) and \(P_2(6)\)?
1. \(\{x \mid x < -4\}\) \(\cup\) \(\{x \mid x > 6\}\)
2. \(\{x \mid x < -4\}\) \(\cup\) \(\{x \mid x > 6\}\)
3. \(\{x \mid x < -4\}\) \(\cup\) \(\{x \mid x > 6\}\)

E. The following ordered pairs = (-1,3)
1. (3,-1)
2. (-1,3)
3. (-1,3) and (3,-1)
4. (-1,3) and (3,-1)
5. (-1,3) and (3,-1)

F. The Cartesian set, \(AxI\), of \(A = (b,B)\) is
1. \((b,B)\)
2. \((b,B),(B,b))\)
3. \((b,B)\)
4. \((b,B),(B,b),(B,b),(B,b)\)
5. \((b,B),(b,B),(B,b),(B,b)\)

G. If \(C = \{-1,0,1\}\) and \(R = \{(0,1),(-1,0)\}\) is a relation in \(C\), then the range of \(R\) is
1. \(R\)
2. \((0,1)\)
3. \((-1,0)\)
4. \((0,1)\)
5. \((0,1)\)
A. If $U = \{1,2,3,4\}$ and $R_1 = \{(x,y) \mid y > x\}$ is a relation on $U$, then $R_1 = \{(1,2),(2,3),(3,4)\}$

B. The graph of the set $\{(x,0) \mid x \text{ is a real number}\}$ is the
1. first quadrant
2. origin
3. $Y$-axis
4. first and fourth quadrants
5. $X$-axis

C. The graph of the set $\{(x,y) \mid x = 3, y \neq 4\}$ is
1. a point
2. a line
3. a rectangular region

D. is the graph of
1. $\{(x,y) \mid x < 2 \text{ and } y < 2\}$
2. $\{(x,0),(0,0),(-2,2)\}$
3. $\{(-2,0),(0,2)\}$
4. $\{(x,y) \mid y = x \text{ and } x \in [-2,2]\}$
5. $\{(x,y) \mid y = x \text{ or } y = -x\}$

E. Assuming the universe of real numbers, the relation $R = \{(x,y) \mid xy \neq 6\}$ is
1. a function
2. not a function
3. the Cartesian set of real numbers
4. an open interval
5. a closed interval

F. is the graph of
1. $\{(x,y) \mid -2 < x < 3 \text{ and } -2 < y < 1\}$
2. $\{(x,0) \mid x \in [-2,3] \text{ or } y \in [1,-2]\}$
3. $\{(x,y) \mid x \in [-2,3] \text{ and } y \in [1,-2]\}$
4. $\{(x,y) \mid -2 < x < 3; U \{(x,y) \mid -2 < y < 1\}\}$
5. $\{(x,y) \mid -2 \leq y \leq 3 \text{ and } -2 \leq x \leq 1\}$

G. Given the relation $R = \{(x,y) \mid x^2 + y^2 = 25\}$. Is $R$ a function?
1. Yes
2. No
A. The domain of the relation \( \{(x,y) | y = \sqrt{25-x^2}\} \) is
1. \( \{x | x^2 = 25\} \)
2. \( \{(x,y) | x^2 < 25\} \)
3. \( \{y | y^2 \leq 25\} \)
4. \( \{x | x^2 \leq 25\} \)
5. \( \{x | x < \sqrt{5}\} \)

B. Given \( P_1(-1,2) \) and \( P_2(-3,3) \), the length of the projection of \( P_1P_2 \) onto the y-axis is
1. 2
2. 3
3. -4

C. The distance between \( P_1(-1,2) \) and \( P_2(-3,3) \) is
1. 3
2. \( \sqrt{13} \)
3. 5
4. \( \sqrt{5} \)

D. The perimeter of triangle \( A(-3,0), B(3,0), C(0,3) \) is
1. \( 6 + 3\sqrt{2} \)
2. 9
3. \( 6(1 + \sqrt{2}) \)
4. 18
5. 12

E. The length of the median from \( A \) to \( BC \) of triangle \( A(0,4), B(-4,0), C(0,0) \) is
1. \( \frac{5}{2} \)
2. \( \frac{7}{2} \)
3. \( \sqrt{10} \)
4. \( \sqrt{8} \)
5. 2

F. Triangle \( A(4,-4), B(4,2), \) and \( C(-4,-2) \)
1. is not a triangle
2. isosceles
3. acute
4. scalene
5. equilateral

G. The coordinates of the point that lies four-ninths of the way from \( P_1(-7,-7) \) to \( P_2(9,9) \) is
1. \( (0,0) \)
2. \( (-3.8,-3.8) \)
3. \( (-3.3) \)
4. \( (5.8,5.8) \)
5. \( (5,5) \)
Quiz #5

A. The points \((-2,1), (\frac{3}{2},-2), (3,-5)\) are collinear.
   1. True
   2. False

B. Given points \((9,3),(4,-2)\) and \((8,6)\), the center of the circle that goes through these three points is
   1. \((4,3)\)
   2. \((6,1)\)
   3. \((8,9)\)

C. The area of triangle \(A(6,-5), B(11,5), C(-5,-5)\) is
   1. 110
   2. 160
   3. 55

D. The circle \((x+1)^2 + (y-3)^2 = 18\) has \(x\) intercepts
   1. 0
   2. 1
   3. 2

E. The center of \((x+2)^2 + (y-7)^2 = 18\) is
   1. \((-2,7)\)
   2. \((2,7)\)
   3. \((0,0)\)

F. The radius of \(2(x+2)^2 + 2(y-4)^2 -16 = 0\) is
   1. 6
   2. \(2\sqrt{2}\)
   3. \(\sqrt{2}\)

G. Given the circle \((x-3)^2 + (y-4)^2 = 16\), the point \((0,0)\) lies
   1. inside the circle
   2. outside the circle
   3. on the circle
A. The graph of \( x^2 + y^2 + 8x - 10y + 50 = 0 \) is a
1. circle
2. point
3. *

B. \( (x,y) \mid x^2 + y^2 = 25 \) and \( x = k \) is
1. \( \{(5,0),(0,5)\} \)
2. \( \{(4,3),(6,-3)\} \)
3. *

C. The equation of a circle whose center is \((2,-3)\) and radius is \(k\) is
1. \( x^2 - 4x + y^2 - 6y - 3 = 0 \)
2. \( (x+2)^2 + (y-3)^2 = 25 \)
3. \( (x-2)^2 + (y+3)^2 + 16 = 0 \)
4. 
5. *

D. The equation of the circle whose diameter has end points \((-3,-2)\) and \((7,-4)\) is
1. \( (x-2)^2 + (y+4)^2 = 100 \)
2. \( (x+2)^2 + (y+4)^2 = 100 \)
3. \( (x+2)^2 + (y-4)^2 = 100 \)
4. 
5. *

E. Given \(A(1,0), B(0,1), C(-5,0), D(-1,-1)\), the area of parallelogram \(ABCD\) is
1. \(8\)
2. \(12\)
3. 
4. *
5. \(ABCD\) is not a parallelogram

F. The \(x\) intercepts of \( x^2 + 8x + y^2 + 4y = 16 = 0 \) are
1. \((-8,0)\)
2. \((0,4),(0,-4)\)
3. *

G. \( (x,y) \mid x^2 + y^2 = 25 \) and \( (x-5)^2 + y^2 = 100 \)
1. \((0,0)\)
2. \((5,0)\)
3. \((0,10)\)
4. *
5. 

Quiz #7

A. The slope and inclination of the line passing through P₁(8,7) and P₂(6,0)
   1. \( m = 1, \alpha = 90^\circ \)  
   2. \( m = -1, \alpha = 135^\circ \)  
   3. \( m = -15^\circ, \alpha = 1 \)

B. The slope of the altitude to side BC of triangle A(-7,5), B(-8,2), C(10,-6) is
   1. \( \frac{1}{9} \)  
   2. -3  
   3. \( \frac{9}{4} \)

C. The equation of a line with slope 2/3 and containing (3,1) is
   1. \( y = \frac{2}{3}x - \frac{2}{3} \)  
   2. \( y = \frac{2}{3}x + 2 \)

D. (-5,3) lies on the perpendicular bisector of the segment whose end points are (-3,-4) and (2,5)
   1. True  
   2. False

E. is the graph of
   1. \( y = -2x + 3 \)  
   2. \( y = -2x - 1 \)  
   3. \( y = 2x + 3 \)  
   4. \( y = 2x - 3 \)  
   5. \( y = 1/2x - 1 \)

F. \( \{(x,y) | x = y\} \cap \{(x,y) | y = 6x\} = \)
   1. (1,7)  
   2. (1/6,1)  
   3. (0,0)  
   4. \( \emptyset \)

G. is the graph of
   1. \( y = -3x - 2 \)  
   2. \( \{(x,y) | x = 1/3x - 2\} \)  
   3. \( y = -3x - 2 \) and \( y < 0 \)  
   4. \( \{(x,y) | y = -1/3x - 2 \text{ and } -3 \leq y \leq 0\} \)  
   5. \( \{(x,y) | x = 1/3x - 2\} \cap \{(x,y) | -3 \leq y \leq 0\} \)
A. Given triangle A(-2,0), B(1,0), and C(0,6). Which of the following equations represents a median?
   1. \( y = 0 \)
   2. \( y = \frac{3}{4}x - 2 \)
   3. \( y = -6x + 6 \)

B. Given the line \( 5x - 12y - 26 = 0 \), (-5,1) is
   1. above the line
   2. below the line
   3. on the line

C. The distance between (3,2) and the line \( x + 4 = 0 \) is
   1. 2
   2. 6
   3. 1

D. The distance between the lines \( 6x + y - 7 = 0 \) and \( x - 3y - 9 = 0 \) is
   1. 10
   2. \( \frac{11}{\sqrt{10}} \)
   3. 2

E. [Diagram]

F. The distance from (7,8) to the circle \( (x-3)^2 + (y-4)^2 - 25 = 0 \) is
   1. (7,8) is inside the circle
   2. \( \frac{5}{2} \)
   3. \( \sqrt{2} - 5 \)

G. The equation of the line tangent to \( (x-3)^2 + (y-4)^2 - 25 = 0 \) at (3,9) is
   1. \( y = 5 \)
   2. \( x + y = 9 \)
   3. \( y = 9 \)
   4. \( x + y = 9 \)
   5. (3,9) is not part of the circle
Quiz #9

A. The angle between two directed lines is never equal to the angle between two undirected lines
1. True  
2. False

B. If L is a directed line from the origin to (3,4) its direction cosines are
1. 3/5, 4/5, 0
2. 6/5, 4/5
3. -3/5, 4/5

C. If L is a directed line from (0,-6) to (-6,0) its direction numbers are
1. -6,6 
2. $\frac{3\sqrt{2}}{2}$, $\frac{3\sqrt{2}}{2}$
3. $\frac{2\sqrt{2}}{2}$, $\frac{1\sqrt{2}}{2}$

D. If (-1,1) and (1,1) are the direction numbers of L₁ and L₂ respectively, the angle between L₁ and L₂ is
1. 0°  
2. 90°  
3. 180°  
4. 45°  
5. cannot be determined

E. The angle between $3x + 3y = 6$ and $3x + y = -3$ is
1. 45°  
2. 30°  
3. arctan 2

F. $x = 7 + 3t; y = 4 - 3t$ are the parametric equations of
1. $x - y = 11$  
2. $y = x + 11$  
3. $y = -3x - 17$

G. $((x,y) | 2x + 3y - 8 = 0) \cap ((x,y) | 6x - 7y + 8 = 0) \cap ((x,y) | 3x - 5y + 7 = 0) =$
1. $(-1,2)$  
2. $(4,0)$  
3. $(2,-1)$
A. The direction cosines from \( P_1(a,0) \) to \( P_2(b,0) \) are
1. \( \frac{a}{\sqrt{a^2+b^2}} \)
2. \( \frac{b}{\sqrt{a^2+b^2}} \)
3. \( \frac{-a}{\sqrt{a^2+b^2}} \)
4. \( \frac{b}{\sqrt{a^2+b^2}} \)
5. \( \frac{-b}{\sqrt{a^2+b^2}} \)

B. \( \{(x,y): x = -5 + 2t, y = 4 + t\} \cap \{(x,y): x + y = 2\} \)
1. \((1,1)\)
2. \((5,-3)\)
3. \((6,-1)\)
4. \((-3,5)\)

C. The vertex of \( y = h \) is
1. \((0,0)\)
2. \((4,0)\)
3. \((0,1/4)\)
4. \((1/4,0)\)
5. \((0,4)\)

D. The directrix of \( y = h \) is
1. \(x = 0\)
2. \(y = 0\)
3. \(x = 4\)
4. \(y = -16\)
5. \(y = -1/16\)

E. The axis of symmetry of \( y = h \) is
1. \(y = h\)
2. \(x = 0\)
3. \(y = 0\)
4. \(x = 1/16\)
5. \(y = -16\)

F. The range and domain of \( y^2 + 6x = 0 \)
1. \([-\infty,0) \) and \( [0,\infty) \)
2. \([-\infty,0) \) and \( (0,\infty) \)
3. \((0,\infty) \) and \( [-\infty,0) \)
4. \([0,\infty) \) and \( (-\infty,0) \)
5. \([0,\infty) \) and \( (-\infty,0) \)

G. \( x^2 + 6x + y^2 + 3 = 0 \) is symmetric with respect to
1. \(x = 0\)
2. the origin
3. \((0,3)\)
4. \((-3,0)\)
5. \(x = 3\)
Quiz #11

A. The length of the latus-rectum of $y = kx^2$ is

1. 16
2. 64
3. 4
4. 1/16

B. is the graph of the parabola

1. $x = -1/2y^2$
2. $x = y^2$
3. $x^2 + y = 3$
4. $y^2 = 2x$
5. $y^2 = 4x$

C. The axis of $y^2 - 10y + 12x + 37 = 0$

1. $y = 5$
2. $x = 3$
3. $x = 0$
4. $y = 0$
5. $y = -9$

D. If $(0,0)$ in the $xOy$ system has coordinates $(-3,5)$ in the $x'Oy'$ system, the origin of $x'Oy'$ has which coordinates in the $xOy$ system?

1. $(0,0)$
2. $(3,-5)$
3. $(5,-3)$
4. $(3,5)$
5. $(-3,5)$

E. If the origin of a new coordinate system is $(-3,5)$, the equation of $(x-1)^2 + y^2 = 9$ relative to this new system is

1. $(x+2)^2 + (y-5)^2 = 9$
2. $(x+4)^2 + (y-5)^2 = 9$
3. $(x+2)^2 + (y+5)^2 = 9$
4. $(x-1)^2 + (y-5)^2 = 9$
5. $(x-1)^2 + (y+5)^2 = 9$

F. If $y = kx - 2x^2$, the coordinates of the highest point on the graph of this equation is

1. $(1,0)$
2. $(2,0)$
3. $(0,0)$
4. $(1,2)$
5. There is no highest value

G. Given the points $(0,-3)$, $(2,-5)$ and $(-2,-1)$, the equation whose graph contains these points is

1. $y + x^2 = 3$
2. $y = x^2 - 3x - 3$
3. $y^2 = x - 3$
A. The graph of \( x = bt + 6, y = 7 \) is
   1. exclusively in quadrants I and II
   2. exclusively in quadrants I and III
   3. exclusively in quadrants I and IV
   4. exclusively in quadrants I, II, and III
   5. in all four quadrants

B. The graph of \( 2y^2 + 2x + 8y - 16 = 0 \) is
   1. exclusively in quadrants I and II
   2. exclusively in quadrants III and IV
   3. exclusively in quadrants I and IV
   4. exclusively in quadrants II and III
   5. in all four quadrants

C. The set of points in a plane with the property that, "the sum of the distance of each point from two fixed points is a constant," is called a
   1. line
   2. plane
   3. circle
   4. parabola
   5. ellipse

D. If the length of the major axis of an ellipse equals the length of its minor axis, the ellipse could be called a
   1. symmetric ellipse
   2. line
   3. circle
   4. hyperbola
   5. parabola

E. The equation of the focal axis of \( x^2 + by^2 = b \) is
   1. \( x = b \)
   2. \( y = b \)
   3. \( x = 0 \)
   4. \( y = 0 \)
   5. \( y = b \)

F. The domain and range of the relation \( R = \{(x,y) | 36x^2 + 25y^2 = 900\} \) are
   1. \([-5;5], [-6;6]\]
   2. \([-5;5], [-6;6]\)
   3. \([-6;6], [-5;5]\)
   4. \([-5;5], [-6;6]\)
   5. \([-6;6], [-5;5]\)

G. The graph of \( 4x^2 - 8x + by^2 + 8y + 6a = 0 \) is
   1. a point
   2. an ellipse with horizontal major axis
   3. an ellipse with vertical major axis
   4. a circle
   5. 
A. The foci of $25x^2 + 16y^2 = 400$ are
1. $(3,0), (-3,0)$
2. $(0,5), (0,-5)$
3. $(4,0), (-4,0)$

B. The length of the latus rectum of $25x^2 + 16y^2 = 400$
1. 8
2. 6
3. 6

C. The center of a hyperbola is the midpoint of its
1. focal axis
2. transverse axis
3. latus rectum

D. A hyperbola is a set of points in a plane with the property that
1. the sum of the distances of each point to two fixed points is a constant
2. the absolute value of the differences of the distances of each point to two fixed points is a constant
3. the distance of each point to a fixed point is a constant
4. each point is equidistant from a point and a line.

E. The domain and range of $R = \{(x,y) \mid 16x^2 - 9y^2 = 144\}$ are
1. $(-\infty, -3) \cup [3;+\infty), (-\infty,-3)$
2. $[-3,3], [-6;6]$
3. $[-3,3] \cup [3;+\infty), [-6;6]$

F. $\{(x,y) \mid x = 0\} \cap \{(x,y) \mid 16x^2 - 9y^2 = 144\} = \{(0,3), (0,-3)\}$

G. The graph of $(x-3)(y-4) = 26$ lies
1. exclusively in quadrants I and III
2. exclusively in quadrants II and IV
3. exclusively in quadrants I, III and IV
4. exclusively in quadrants I, II, and III
5. in all four quadrants
Computerized Analytic Geometry

Quiz #14  7 May 71

A. A plane parallel to one element of a double-napped right circular cone that cuts every other element of the cone forms
1. a parabola
2. a hyperbola
3. intersecting lines
4. an ellipse
5. a circle

B. A plane perpendicular to the axis of a cone that intersects the cone at a point other than its vertex forms
1. a parabola
2. a hyperbola
3. intersecting lines
4. an ellipse
5. a circle

C. Given the equation $Ax^2 + Cy^2 + Dx + Ey + F = 0$, if $A \cdot C > 0$, the graph of the equation cannot be
1. a circle
2. an ellipse
3. a parabola
4. a point

D. Given the equation $Ax^2 + Cy^2 + Dx + Ey + F = 0$, which of the following set of relationships must exist to guarantee a circle?
1. $A = C$, $F < \frac{A}{D+E}$
2. $AC > 0$, $F > D+E$
3. $A = C = 0$, $F < 0$

E. The equation of the set of points equidistant from (-3,0) and (3,0) is
1. $x^2 + y^2 = 9$
2. $y = 0$
3. $5y^2 - x^2 = 5$
4. $5y^2 + x^2 = 5$

F. If a hyperbola had the equation $x^2 - 5y^2 = 5$ in xOy system, its equation in the $x^*O^*y^*$ system where $x^* = 3$ and $y^* = -5$ is
1. $(x-3)^2 - 5(y+5)^2 = 5$
2. $x^2 - 5y^2 = 5$
3. $3x^2 + 10y^2 = 5$
4. $(x+3)^2 - 5(y-5)^2 = 5$

G. $\{(x,y) \mid 9x^2 + 16y^2 = 144\}$
1. $\{(4,0), (4,0)\}$
2. $\{(4,0), (-4,0), (0,3)\}$
3. $\{(0,3), (0,-3), (0,3), (0,-3)\}$
4. $\{(0,3), (0,-3)\}$
5. $\{(0,3), (0,-3)\}$
A. To generate a directed light from a automobile headlight, the light source is placed
1. at either focus of an ellipse
2. at the center of a circle
3. at the focus of a parabola
4. at the vertex of a cone
5. at the intersection of two parallel lines

B. The equation of the set of points equidistant from the point (1, -3) and the line whose equation is \( y = 5 = 0 \) is
1. \( x^2 = -16y \)
2. \((x-1)^2 = -4(y-1)\)
3. \((x+1)^2 = 16(y+4)\)

C. If the graph of equation \( x^2 + 6y = 36 \) is rotated 90° clockwise, its new equation is
1. \( (y^*)^2 + 6x^* = 36 \)
2. \((x^*)^2 + 6y^* = 36 \)
3. \((y^*)^2 - 1/6x^* = 36 \)

D. If the graph of equation \( x^2 + 4(y-2)^2 = 16 \) is rotated 180°, its new equation is
1. \( (x^*)^2 + 4(y^*-2)^2 = 16 \)
2. \((x^*)^2 + 4(y^*-2)^2 = 16 \)
3. \((y^*)^2 - 1/4x^* = 36 \)

E. In the general equation \( Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0 \) if \( A=C \), then the graph of the equation is a
1. circle
2. ellipse
3. hyperbola
4. parabola
5. cannot be determined

F. In the general equation \( Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0 \) if \( A=C \) and \( B\neq0 \), then the angle of rotation to eliminate the \( xy \) term is
1. \( \arctan \frac{B-A}{2A} \)
2. \( 90° \)
3. \( 180° \)
4. \( 0° \)

G. If the graph of the general equation \( Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0 \) is rotated so as to eliminate the \( xy \) term, which of the terms do not change
1. \( A \) and \( D \)
2. \( D \) and \( F \)
3. \( F \)
4. \( A \)
5. All terms may change
Quiz #16

The polar coordinates of \( P_1 \) are probably

1. \([3,3, -30^\circ]\)
2. \([3, \frac{\pi}{3}]\)
3. \([2,3]\)
4. \([3, \frac{\pi}{6}]\)
5. \([3, \frac{\pi}{4}]\)

12° expressed in radians is

1. \(-\frac{1}{30} \pi\)
2. \(\frac{\pi}{180}\)
3. \(\frac{\pi}{180} \pi\)

\( x^2 + y^2 = 9 \) in rectangular can be converted to ___ in polar coordinates

1. \( \theta = 3 \)
2. \( r = 3 \)
3. \( r = 9 \)
4. \( r = -3 \)
5. \( r = 1 + 2\cos \theta \)

The domain and range of \( R = \{(r, \theta) \mid r = 2(1-\cos \theta)\} \) are

1. \([0; 1], [0; \pi]\)
2. \([-1; 1], [0; \pi]\)
3. \([0; 2], [-\pi; \pi]\)
4. \([0; 2], [0; 2\pi]\)
5. \([0; 2], [0; 2\pi]\)

The graph of the equation \( r = \sin 2\theta \) is

1. a cardioid
2. symmetric to the origin
3. symmetric to the x-axis only
4. symmetric to the y-axis only
5. a circle

The graph of \( r^2 = 4\sin 2\theta \) is a

1. 4-leaved rose
2. circle
3. cardioid

The graph of \( \{(r, \theta) \mid 0 < r < 2, -\frac{\pi}{2} < \theta < \frac{\pi}{2}\} \) lies

1. completely in quadrant I
2. completely in quadrant II
3. completely in quadrants I and II
4. completely in quadrants I and IV
5. in all four quadrants
A. The rectangular coordinates of the point P(6,7) are
1. (6,0)
2. (0,6)
3. (-6,0)
4. (-6,6)
5. (0,-6)

B. The curve \( r = \frac{a}{\sqrt{2-cos\theta}} \) in polar coordinates converts to a _______ in rectangular coordinates
1. circle
2. parabola
3. ellipse
4. hyperbola
5. point

C. A curve has the equation \( x^2 + 4y^2 = 4 \) in rectangular coordinates; its equation in polar coordinates is
1. \( r = 2 \)
2. \( r^2 (\cos^2 \theta + \sin^2 \theta) = 4 \)
3. \( r (\cos \theta + \sin \theta) = 2 \)
4. \( r^2 = 4 \)
5. \( r^2 (\sin^2 \theta + \cos^2 \theta) = 4 \)

D. Two different equations in polar coordinates may have the same graph.
1. True
2. False

E. A curve has the equation \( r(2\cos \theta + 3\sin \theta = 5 \) in polar coordinates; its equation in rectangular coordinates is
1. \( 2x + 3y = 5 \)
2. \( 2y + 3x = 5 \)
3. \( x + y = 5 \)
4. \( 2x + 3y = 5 \)
5. \( 2x^2 + 3y^2 = 5 \)

F. The graph of \( r = 5\sin \theta \) is a
1. 2-leaf rose
2. 4-leaf rose
3. 8-leaf rose
4. 8-leaf rose
5. lemniscate

G. The domain and range of \( R = \{ (r, \theta) \mid r = \frac{8}{2 \cos \theta} \} \) are
1. \( \left[ \frac{4}{3}, \infty \right) \), \((-\infty, 0)\)
2. \( \left[ -\frac{2\sqrt{2}}{3}, \frac{2\sqrt{2}}{3} \right) \), \((-\infty, 0)\)
3. \( \left[ -\infty, \frac{\pi}{8} \right) \), \(\left( \frac{\pi}{2}, \infty \right)\) respectively
Quiz #16

A. Is the graph of
1. \( r = 2 \cos \theta \)
2. \( r = 2 \sin \theta \)
3. \( r^2 = 16 \cos \theta \)
4. \( r = 4 \sin \theta \)
5. \( r = 2 \sin \theta \)

3. The graph of \( r \cos \theta = k \) is a
1. point
2. line
3. circle

4. The set of points in a plane with the property that the distance between each point and a fixed point is equal to \( "e" \) times the distance between the given point and a fixed line, \( e > 0 \), is
1. a parabola
2. an ellipse
3. a hyperbola
4. a circle
5. a conic section

6. The graph of \( r = \frac{8}{2-\cos \theta} \)
1. hyperbola
2. line
3. circle
4. parabola
5. ellipse

7. The graph of \( r = \frac{8}{1-\cos \theta} \) lies
1. exclusively in quadrants I and II
2. exclusively in quadrants II and III
3. exclusively in quadrants III and IV
4. exclusively in quadrants I and IV
5. in all four quadrants

F. The equations of the asymptotes of \( r = \frac{8}{2 \sin \theta} \) are
1. \( r \sin \theta = 0, r \sin \theta = 2 \)
2. \( r \sin \theta = 1, r \sin \theta = 2 \)
3. \( \theta = 0, \theta = \frac{\pi}{2} \)
4. \( \theta = \frac{\pi}{2}, \theta = \frac{\pi}{2} \)
5. \( \theta = 0, \theta = \frac{\pi}{2} \)

3. \( \{ (r, \theta) \mid \sin \theta = 1 \} \) \( \{ (r, \theta) \mid \theta = \frac{\pi}{2} \} \)
1. \( \{(2, \frac{\pi}{2})\} \)
2. \( \{(2, \frac{\pi}{2}), (2, \frac{\pi}{2})\} \)
3. \( \{(2, \frac{\pi}{2}), (-2, \frac{\pi}{2})\} \)
### A.2 Quiz item analysis

<table>
<thead>
<tr>
<th>Quiz Number</th>
<th>Number missing quiz item</th>
<th>Number of Students taking quiz</th>
<th>Student Average(^a)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9 8 14 16 4 5 13</td>
<td>18</td>
<td>3.2</td>
</tr>
<tr>
<td>2</td>
<td>10 2 6 8 6 0 9</td>
<td>17</td>
<td>4.6</td>
</tr>
<tr>
<td>3</td>
<td>0 0 5 3 5 9 3</td>
<td>18</td>
<td>5.6</td>
</tr>
<tr>
<td>4</td>
<td>10 4 0 5 2 10 5</td>
<td>17</td>
<td>4.9</td>
</tr>
<tr>
<td>5</td>
<td>3 7 4 2 1 1 4</td>
<td>18</td>
<td>5.8</td>
</tr>
<tr>
<td>6</td>
<td>3 3 4 6 13 11 6</td>
<td>18</td>
<td>4.4</td>
</tr>
<tr>
<td>7</td>
<td>4 8 1 9 0 3 9</td>
<td>18</td>
<td>5.1</td>
</tr>
<tr>
<td>8</td>
<td>8 7 2 1 8 9 3</td>
<td>18</td>
<td>4.9</td>
</tr>
<tr>
<td>9</td>
<td>1 10 11 7 16 7 5</td>
<td>18</td>
<td>3.8</td>
</tr>
<tr>
<td>10</td>
<td>10 4 1 12 5 12 10</td>
<td>18</td>
<td>4.0</td>
</tr>
<tr>
<td>11</td>
<td>8 2 9 4 13 7 5</td>
<td>18</td>
<td>4.3</td>
</tr>
<tr>
<td>12,</td>
<td>6 10 0 0 8 4 4</td>
<td>18</td>
<td>5.2</td>
</tr>
<tr>
<td>13</td>
<td>2 7 8 1 2 4 8</td>
<td>15</td>
<td>4.9</td>
</tr>
<tr>
<td>14</td>
<td>10 1 10 13 2 7 3</td>
<td>18</td>
<td>4.4</td>
</tr>
<tr>
<td>15</td>
<td>9 10 8 13 7 8 9</td>
<td>17</td>
<td>3.2</td>
</tr>
<tr>
<td>16</td>
<td>2 1 1 13 4 13 1</td>
<td>17</td>
<td>4.9</td>
</tr>
<tr>
<td>17</td>
<td>4 9 7 1 7 13 9</td>
<td>17</td>
<td>4.1</td>
</tr>
<tr>
<td>18</td>
<td>7 9 14 4 7 14 8</td>
<td>15</td>
<td>2.8</td>
</tr>
</tbody>
</table>

\(^a\) Maximum score = 7
B.1 Homework assignments

Computerized Analytic Geometry

Programs due 20 Apr 71

1. INPUT: X1 = number of elements in set A
   X2 = number of elements in set B
   A = array of elements in set A
   B = array of elements in set B

   OUTPUT: E = 1 if A=B
            = 0 if A^B
   N3 = number of elements in set A∩B
   N4 = number of elements in set AUB
   C = array of elements in set AUB

   STATEMENT NUMBERS: >=1000 and <1200

   NOTE: For this program and all subsequent programs assume all arrays have been properly dimensioned and that an END statement is unnecessary. Terminate all programs with a GO to 9000. All variables ending in a 9 (i.e. P9) and the U and V arrays are not to be used by the student.

2. INPUT: X1 = coordinate of point P1 in a one-dimensional coordinate system
   X2 = coordinate of point P2
   X3 = coordinate of point P3

   OUTPUT: D = the minimum of all possible directed distances
            E = the maximum of all possible undirected distances

   STATEMENT NUMBERS: >=1200 and <1400

Programs due 21 Apr 71

3. INPUT: X1 = number of elements in A
   A = array of elements in set A

   OUTPUT: X2 = number of elements in AxA
            B = array of left members of ordered pairs of AxA
            C = array of right members of ordered pairs of AxA

   STATEMENT NUMBERS: >=1400 and <1600

4. INPUT: X1 = number of elements in set A
   A = array of elements in set A
   FNA(X) = arbitrary predefined function

   OUTPUT: X2 = number of elements of R = \{(x,y) \mid x \in A \text{ and } y = FNA(X)\} where
            R is a relation in A, I, A, \ldots, I, A
   B = array of left members of ordered pairs of R
   C = array of right members of ordered pairs of R

   STATEMENT NUMBERS: >=1600 and <1800
Programs due 22 Apr 71

5. INPUT:  
   $X_1 =$ number of elements in $A$  
   $A =$ array of elements in set $A$  
   $N_2 =$ number of elements of $X_1$, a relation in $A$  
   $B =$ array of left members of ordered pairs of $R$  
   $C =$ array of right members of ordered pairs of $R$  

   OUTPUT:  
   $F = 1$ if $R$ is a function  
   $= 0$ if $R$ is not a function  

   STATEMENT NUMBERS: >1800 and <2000

6. INPUT: $X_1,Y_1$, $X_2,Y_2$, $X_3,Y_3$  
   coordinates of the vertices of a rectangle whose sides  
   are parallel to the $X$ and $Y$ axis  

   OUTPUT: $X_4,Y_4 =$ coordinates of the four vertex of the rectangle  
   $A =$ the area of the rectangle  

   STATEMENT NUMBERS: >2000 and <2200

Programs due 23 Apr 71

7. INPUT: $X_1,Y_1$, $X_2,Y_2$  
   coordinates of the end points of a segment  

   OUTPUT:  
   $X =$ length of the projection of the segment on the $X$-axis  
   $Y =$ length of the projection of the segment on the $Y$-axis  
   $D =$ length of the segment  

   STATEMENT NUMBERS: >2200 and <2400

8. INPUT: $X_1,Y_1$, $X_2,Y_2$, $X_3,Y_3$  
   coordinates of the vertices of a triangle  

   OUTPUT  
   $T = -1$ if the points are not collinear and do not form a triangle  
   $= 0$ if the points are collinear  
   $= 1$ if the triangle is scalene  
   $= 2$ if the triangle is isosceles  
   $= 3$ if the triangle is equilateral  
   $P =$ the perimeter of the triangle  

   STATEMENT NUMBERS: >2400 and <2600
Progress due 26 Apr 71

9. INPUT: \((X_1, Y_1), (X_2, Y_2), (X_3, Y_3)\) coordinates of the vertices of a triangle

OUTPUT: 
- \(M_1 = \) length of median from \((X_1, Y_1)\)
- \(M_2 = \) length of median from \((X_2, Y_2)\)
- \(M_3 = \) length of median from \((X_3, Y_3)\)

STATEMENT NUMBERS: \(>2600\) and \(<2800\)

HINT: Use Prob 8 to first determine if you have a triangle

10. INPUT: \((X_1, Y_1), (X_2, Y_2), (X_3, Y_3), (X_4, Y_4)\) coordinates of the vertices of a quadrilateral (Given in proper order)

OUTPUT: 
- \(Q = 1\) if the quadrilateral is a square
- \(Q = 2\) if the quadrilateral is a rhombus
- \(Q = 3\) if the quadrilateral is a rectangle
- \(Q = 4\) if the quadrilateral is a parallelogram
- \(Q = 5\) if the quadrilateral is an isosceles trapezoid
- \(Q = 6\) if the quadrilateral is a kite
- \(Q = 0\) if the quadrilateral is none of these

\(P = \) the perimeter of the quadrilateral

STATEMENT NUMBERS: \(>2800\) and \(<3000\)

NOTE: Set \(Q\) to the most definitive of the choices; i.e., a square is also a rectangle but \(Q\) should be set to 1 for a square

Program due 27 Apr 71

11. INPUT: \(A, C, D, E, F\) which are coefficients of the equation \(Ax^2 + Cy^2 + Dx + Ey + F = 0\)

OUTPUT: 
- \(C = 1\) if the graph of the equation is a circle
- \(0\) if the graph of the equation is not a circle

\(C1, C2 = \) coordinates of the center of the circle
\(R = \) the radius of the circle
\(N = \) the number of elements in \(X\)
\(X = \) the array of \(X\) intercepts

STATEMENT NUMBERS: \(>3000\) and \(<3200\)

12. INPUT: \(A, C, D, E, F\) which are coefficients of the equation \(Ax^2 + Cy^2 + Dx + Ey + F = 0\) and \((x, y) = \) coordinate of an arbitrary point

OUTPUT: 
- \(P = -1\) if \((x, y)\) is inside the circle
- \(P = 0\) if \((x, y)\) is on the circle
- \(P = 1\) if \((x, y)\) is outside the circle
- \(P = 2\) if the graph of the equation is not a circle

STATEMENT NUMBERS: \(>3200\) and \(<3400\)

HINT: Use Program 11
13. **INPUT:** \(X_1, Y_1\) coordinates of three points
\(X_2, Y_2\)
\(X_3, Y_3\)

**OUTPUT:**
- \(L = 1\) if the points are collinear
- \(L = 0\) if the points are not collinear

**STATEMENT NUMBERS:** >3400 and <3600

**NOTE:** Do not use the distance formula

14. **INPUT:** \((X_1, Y_1)\) coordinate of vertex A of triangle ABC
\((X_2, Y_2)\) coordinate of vertex B
\((X_3, Y_3)\) coordinate of vertex C

**OUTPUT:**
- \(M_1, B_1 = \text{coefficients of } y = M_1x + B_1, \text{ the equation of the line containing the median from point A to side BC}\)
- \(M_2, B_2 = \text{coefficients of } y = M_2x + B_2, \text{ the equation of the line containing the altitude from point A to side BC}\)
- \(M_3, B_3 = \text{coefficients of } y = M_3x + B_3, \text{ the equation of the line containing the perpendicular bisector of side BC}\)

**STATEMENT NUMBERS:** >3600 and <3800

Programs due 28 Apr 71

15. **INPUT:** \(D, E, F\) which are coefficients of the equation \(x^2 + y^2 + Dx + Ey + F = 0\)
\(x, y\) = coordinates of a point of the circle

**OUTPUT:**
- \(P = 0\) if \((x, y)\) is not on the circle
- \(M, B = \text{coefficients of } y = Mx + B, \text{ the equation of the tangent to the circle at } (X_1, Y_1)\)

**STATEMENT NUMBERS:** >3800 <4000

16. **INPUT:** \(A_1, B_1, C_1\) the coefficients of \(A_1x + B_1y + C_1 = 0\)
\(A_2, B_2, C_2\) the coefficients of \(A_2x + B_2y + C_2 = 0\)

**OUTPUT:**
- \(P = 1\) if the lines are parallel
- \(P = 0\) if the lines are not parallel
- \((x, y) = \text{the coordinates of the intersection}\)
- \(D = \text{the distance between the lines if they are parallel}\)

**STATEMENT NUMBERS:** >4000 <4200
17. **INPUT:** \((x_1, y_1), (x_2, y_2), (x_3, y_3)\) coordinates of the vertices of a triangle \(ABC\), respectively

**OUTPUT:**
- \(A\) = measure of angle \(A\) in degrees
- \(B\) = measure of angle \(B\) in degrees
- \(C\) = measure of angle \(C\) in degrees
- \(T = 0\) if triangle is acute
- \(T = 1\) if triangle is obtuse

**STATEMENT NUMBERS:** >200 and <400

18. **INPUT:**
- \(m_1, b_1 = \text{coefficients of } y = m_1x + b_1\)
- \(m_2, b_2 = \text{coefficients of } y = m_2x + b_2\)
- \(m_3, b_3 = \text{coefficients of } y = m_3x + b_3\)

**OUTPUT:**
- \(S = 0\) if the lines are collinear
- \(S = 1\) if the lines are parallel
- \(S = 2\) if the lines are concurrent
- \(S = 3\) if the lines intersect in two points
- \(S = 4\) if the lines intersect in a triangle

**STATEMENT NUMBERS:** >400 and <600

19. **INPUT:** \(A, B, C, D, E = \text{coefficients of the equation } Ax^2 + By^2 + Cx + Dy + E = 0\)

**OUTPUT:**
- \(N = 1\) if the graph of the equation is not a parabola
- \(X_1, Y_1 = \text{coordinates of the focus}\)
- \(A_1, A_2, A_3 = \text{coefficients of } A_1x + A_2y + A_3 = 0, \text{the equation of the directrix}\)
- \(B_1, B_2, B_3 = \text{coefficients of } B_1x + B_2y + B_3 = 0, \text{the equation of the axis}\)
- \(L = \text{length of the latus rectum}\)

**STATEMENT NUMBERS:** >400 and <600

**NOTE:** You can assume that either \(C = 0\) or \(D = 0\), but not both = 0. Also that \(E = 0\).

20. **INPUT:** \((x_1, y_1), (x_2, y_2), (x_3, y_3)\) coordinates of three arbitrary points

**OUTPUT:**
- \(A, B, C = \text{coefficients of } y = Ax^2 + Bx + C, \text{a parabola containing all three of these points}\)
- \(P = 0\) if the points are collinear

**STATEMENT NUMBERS:** >400 and <5000

**HINT:** Use program 8 as a subroutine to determine if points are collinear. This problem can be reduced to finding the solution of 3 equations in 3 variables.
Programs due 4 May 71

21. Same as program 19 except that C, D, and E can be nonzero. Use statement numbers >5000 and <5020

22. INPUT: \( X1, Y1 \) = the coefficients of the focus of a parabola 
\( X2, Y2 \) = the coefficients of the vertex of a parabola

OUTPUT: \( A, B, C, D, E \) the coefficients of \( Ax^2 + By^2 + Cx + Dy + E = 0 \) the parabola with \((X1, Y1)\) as focus and \((X2, Y2)\) as vertex

STATEMENT NUMBERS: >5200 and <5400

NOTE: Assume the axis of the parabola is parallel to the X or Y axis.

Programs due 5 May 71

23. INPUT: \( A, B, C, D, E \) = coefficients of the equation \( Ax^2 + By^2 + Cx + Dy + E = 0 \)

OUTPUT: \( k = 1 \) if the graph of the equation is not an ellipse
\( X1, Y1 \) = the coordinates of the focii
\( X2, Y2 \) = length of the major axis
\( L2 \) = length of the minor axis

STATEMENT NUMBERS: >5400 and <5600

24. INPUT: \( X1, Y1 \) = coordinates of the focii of an ellipse 
\( X2, Y2 \) = coordinates of a point on the ellipse

OUTPUT: \( A, C, D, E, F \) the coefficients of the equation \( Ax^2 + Cy^2 + Dx + Ey + F = 0 \) which is the generalized equation of an ellipse whose axes are parallel to the X and Y-axis.

STATEMENT NUMBERS: >5600 and <5800

NOTE: Check to be sure that the axes are parallel to the X and Y-axis.
Programs due 6 May 71

25. INPUT: \( A,C,D,E,F \) the coefficients of the equation \( Ax^2 + Cy^2 + Dx + Ey + F = 0 \)

OUTPUT: \( \gamma = 1 \) if the graph of the equation is not a hyperbola

\( X_1,Y_1 \) the coordinates of the foci
\( X_2,Y_2 \) the coordinates of its center
\( L_1 \) the length of the transverse axis
\( L_2 \) the length of the latus rectum.

STATEMENT NUMBERS: >5500 and <6000

26. INPUT: \( X_1,Y_1 \) coordinates of the foci of a hyperbola
\( X_2,Y_2 \) coordinates of a point on the hyperbola

OUTPUT: \( A,C,D,E,F \) the coefficients of the equation \( Ax^2 + Cy^2 + Dx + Ey + F = 0 \) which is the generalized equation of a hyperbola whose axis are parallel to the X and Y axis.

STATEMENT NUMBERS: >5500 and <6000

NOTE: Check to be sure that the axes are parallel to the X and Y-axis.

Programs due 7 May 71

27. INPUT: \( C_1,C_2 \) constants of \( X=C_1 \), \( Y=C_2 \) the asymptotes of a rectangular hyperbola
\( X_1,Y_1 \) coordinates of a point on the hyperbola

OUTPUT: \( D,E,F \) the coefficients of equation \( xy + Dx + Ey + F = 0 \) which is the generalized equation for a rectangular hyperbola.

STATEMENT NUMBERS: >6000 and <6200

28. INPUT: \( A,C,D,E,F \), the coefficients of the equation \( Ax^2 + Cy^2 + Dx + Ey + F = 0 \)

OUTPUT: \( G = 0 \) if the graph of the equation is \( 0 \)
\( G = 1 \) if the graph of the equation is a point
\( G = 2 \) if the graph of the equation is a circle
\( G = 3 \) if the graph of the equation is a parabola
\( G = 4 \) if the graph of the equation is an ellipse
\( G = 5 \) if the graph of the equation is a hyperbola
\( G = 6 \) if the graph of the equation is none of these.

STATEMENT NUMBERS: >6200 and <6400
Programs due 10 May 71

29. INPUT: A, B, C, D, E, F the coefficients of the equation
       \[ Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0 \]

OUTPUT: (M1, M2, B1, B2) the coefficients of
         \[ y = M1x + B1 \]
         \[ y = M2x + B2 \]

the equations in the xOy system of the coordinate axes of
x"O'y" system after a rotation to eliminate the xy term and
a translation to eliminate the x and y term.

STATEMENT NUMBERS: >6400 and <6600

30. INPUT: A, B, C, D, E, F the coefficients of the equation
       \[ Ax^2 + Bxy = Cy^2 + Dx + Ey + F = 0 \]

OUTPUT: G = 0 if the graph of the equation = ∅
G = 1 if the graph of the equation is a point
G = 2 if the graph of the equation is a circle
G = 3 if the graph of the equation is a parabola
G = 4 if the graph of the equation is an ellipse
G = 5 if the graph of the equation is a hyperbola
G = 6 if the graph of the equation is none of these

STATEMENT NUMBERS: >6600 and <6800

HINT: Use program 28 as a subprogram

Programs due 11 May 71

31. INPUT: N = the number of elements in an array of polar coordinates (r,θ)
       X = the array of left elements of the ordered pairs
       T = the array of right elements of the ordered pairs
       FNA = an arbitrary function which represents a curve in polar coordinates

OUTPUT: N2 = the number of elements of the input array which satisfy r = FNA(θ)
N2 = the array of r values satisfying r = FNA(θ)
N2 = the array of θ values satisfying r = FNA(θ)

STATEMENT NUMBERS: >6800 <7000

32. INPUT: A, B, C, D, E constants of the equation r + A sinθ + B cosθ + E

OUTPUT: P = the maximum value of |r|
P = 0 if the graph of the equation = ∅
P = 1 if the graph of the equation is a CARDIOD
P = 2 if the graph of the equation is a ROSE
P = 3 if the graph of the equation is a LIMACON
N = the number of leaves if P = 2

STATEMENT NUMBERS: >7000 and <7200
Programs due 12 May 71

33. INPUT: R, T the polar coordinates of an arbitrary point
OUTPUT: X, Y the rectangular coordinates the same point
STATEMENT RUGERS: >7000 and <7000

34. INPUT: A, B, C, D, E, F the coefficients of the equation
\[ Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0 \]
A1, B1, C1, D1, E1, F1 the coefficients of \( r = A1\sin B1\theta + C1\cos D1\theta + E1 \)
A2, B2, C2, D2, E2, F2 the coefficients of \( r = A2\sin B2\theta + C2\cos D2\theta + E2 \)
T1 = tolerance level
OUTPUT: \( X = Y \) if the two equations are not equivalent
STATEMENT RUGERS: >7000 and <7000
HINT: Consider the respective coordinates of two points equal in the xOy coordinate system if their difference is less than T, the tolerance level.

Programs due 13 May 71

35. INPUT: D, E, F the coefficients of the equation \( x^2 + y^2 + Dx + Ey + F = 0 \), a circle in the xOy coordinate system
OUTPUT: A, B, C the coefficients of the equation \( r^2 - A\cos(O-B) + C = 0 \) a circle in the polar coordinate system
STATEMENT RUGERS: >7000 and <7000

36. INPUT: A1, B1, C1, D1, E1, F1 the coefficients of \( r = A1\sin B1\theta + C1\cos D1\theta + E1 \)
A2, B2, C2, D2, E2, F2 the coefficients of \( r = A2\sin B2\theta + C2\cos D2\theta + E2 \)
OUTPUT: \( C = 0 \) if its graph is a line
\( C = 1 \) if its graph is a circle
\( C = 2 \) if its graph is a parabola
\( C = 3 \) if its graph is an ellipse
\( C = 4 \) if its graph is a hyperbola
\( C = 5 \) if its graph is none of these
STATEMENT RUGERS: >7000 and <7000
Programs due 1st May 71

37. INPUT: \((X_1,Y_1,Z_1)\), \((X_2,Y_2,Z_2)\), \((X_3,Y_3,Z_3)\) the coordinates of a triangle in 3-space

OUTPUT: \(M_1, M_2, M_3\) = the lengths of the three medians of the triangle

STATEMENT NUMBERS: >8000 and <8200

38. INPUT: \(A, B, C, D, E, F, G\) the coefficients of the equation
\[Ax^2 + By^2 + Cz^2 + Dx + Ey + Fz + G = 0\]

OUTPUT: \(S = 0\) if the graph of the equation is not a sphere,
\(R = \) the radius of the sphere,
\((X_1,Y_1,Z_1)\), \((X_2,Y_2,Z_2)\), \((X_3,Y_3,Z_3)\), the coordinates of the center of the sphere

STATEMENT NUMBERS: >8200 and <8400
### B.2 Table of check program characteristics

<table>
<thead>
<tr>
<th>Program Number</th>
<th>No. of BASIC Statements in Sample Solution</th>
<th>No. of BASIC Statements in Check Program</th>
<th>Data Type</th>
<th>No. of Data Sets in Check Program</th>
<th>Writing Time (^a) (in Minutes)</th>
<th>Debugging Time (^b) (in Minutes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20</td>
<td>86</td>
<td>R</td>
<td>5</td>
<td>118</td>
<td>118</td>
</tr>
<tr>
<td>2</td>
<td>15</td>
<td>52</td>
<td>R</td>
<td>5</td>
<td>118</td>
<td>118</td>
</tr>
<tr>
<td>3</td>
<td>9</td>
<td>67</td>
<td>R</td>
<td>5</td>
<td>118</td>
<td>118</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>74</td>
<td>R</td>
<td>5</td>
<td>175</td>
<td>175</td>
</tr>
<tr>
<td>5</td>
<td>16</td>
<td>63</td>
<td>R</td>
<td>5</td>
<td>118</td>
<td>118</td>
</tr>
<tr>
<td>6</td>
<td>21</td>
<td>43</td>
<td>F</td>
<td>5</td>
<td>106</td>
<td>106</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>44</td>
<td>R</td>
<td>5</td>
<td>106</td>
<td>106</td>
</tr>
<tr>
<td>8</td>
<td>19</td>
<td>47</td>
<td>F</td>
<td>5</td>
<td>106</td>
<td>106</td>
</tr>
<tr>
<td>9</td>
<td>4</td>
<td>48</td>
<td>R</td>
<td>5</td>
<td>116</td>
<td>116</td>
</tr>
<tr>
<td>10</td>
<td>28</td>
<td>49</td>
<td>F</td>
<td>7</td>
<td>106</td>
<td>106</td>
</tr>
<tr>
<td>11</td>
<td>23</td>
<td>82</td>
<td>R</td>
<td>5</td>
<td>176</td>
<td>176</td>
</tr>
<tr>
<td>12</td>
<td>13</td>
<td>56</td>
<td>R</td>
<td>5</td>
<td>88</td>
<td>88</td>
</tr>
<tr>
<td>13</td>
<td>3</td>
<td>48</td>
<td>R</td>
<td>5</td>
<td>38</td>
<td>38</td>
</tr>
<tr>
<td>14</td>
<td>7</td>
<td>58</td>
<td>R</td>
<td>5</td>
<td>38</td>
<td>38</td>
</tr>
<tr>
<td>15</td>
<td>8</td>
<td>52</td>
<td>R</td>
<td>5</td>
<td>48</td>
<td>48</td>
</tr>
<tr>
<td>16</td>
<td>9</td>
<td>58</td>
<td>R</td>
<td>5</td>
<td>63</td>
<td>63</td>
</tr>
<tr>
<td>17</td>
<td>21</td>
<td>68</td>
<td>R</td>
<td>5</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>18</td>
<td>16</td>
<td>43</td>
<td>F</td>
<td>5</td>
<td>33</td>
<td>33</td>
</tr>
<tr>
<td>19</td>
<td>27</td>
<td>60</td>
<td>F</td>
<td>5</td>
<td>30</td>
<td>60</td>
</tr>
<tr>
<td>20</td>
<td>8</td>
<td>61</td>
<td>R</td>
<td>5</td>
<td>20</td>
<td>55</td>
</tr>
<tr>
<td>21</td>
<td>29</td>
<td>61</td>
<td>F</td>
<td>5</td>
<td>65</td>
<td>48</td>
</tr>
<tr>
<td>22</td>
<td>15</td>
<td>52</td>
<td>F</td>
<td>5</td>
<td>40</td>
<td>58</td>
</tr>
<tr>
<td>23</td>
<td>20</td>
<td>54</td>
<td>F</td>
<td>5</td>
<td>65</td>
<td>48</td>
</tr>
<tr>
<td>24</td>
<td>25</td>
<td>46</td>
<td>F</td>
<td>5</td>
<td>60</td>
<td>33</td>
</tr>
<tr>
<td>25</td>
<td>23</td>
<td>63</td>
<td>F</td>
<td>5</td>
<td>80</td>
<td>33</td>
</tr>
<tr>
<td>26</td>
<td>25</td>
<td>46</td>
<td>F</td>
<td>5</td>
<td>55</td>
<td>23</td>
</tr>
<tr>
<td>27</td>
<td>5</td>
<td>44</td>
<td>R</td>
<td>5</td>
<td>15</td>
<td>18</td>
</tr>
<tr>
<td>28</td>
<td>4</td>
<td>40</td>
<td>F</td>
<td>7</td>
<td>65</td>
<td>21</td>
</tr>
<tr>
<td>29</td>
<td>15</td>
<td>63</td>
<td>R</td>
<td>5</td>
<td>25</td>
<td>41</td>
</tr>
<tr>
<td>30</td>
<td>21</td>
<td>41</td>
<td>F</td>
<td>7</td>
<td>25</td>
<td>16</td>
</tr>
<tr>
<td>31</td>
<td>8</td>
<td>69</td>
<td>R</td>
<td>5</td>
<td>35</td>
<td>26</td>
</tr>
<tr>
<td>32</td>
<td>24</td>
<td>45</td>
<td>F</td>
<td>5</td>
<td>20</td>
<td>16</td>
</tr>
<tr>
<td>33</td>
<td>4</td>
<td>40</td>
<td>R</td>
<td>5</td>
<td>15</td>
<td>16</td>
</tr>
<tr>
<td>34</td>
<td>11</td>
<td>56</td>
<td>R</td>
<td>5</td>
<td>20</td>
<td>21</td>
</tr>
<tr>
<td>35</td>
<td>4</td>
<td>42</td>
<td>F</td>
<td>5</td>
<td>25</td>
<td>16</td>
</tr>
<tr>
<td>36</td>
<td>18</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>13</td>
<td>72</td>
<td>R</td>
<td>5</td>
<td>15</td>
<td>31</td>
</tr>
<tr>
<td>38</td>
<td>8</td>
<td>57</td>
<td>R</td>
<td>5</td>
<td>10</td>
<td>27</td>
</tr>
</tbody>
</table>

**Average:**

|                  | 15 | 55 | 36 | 64 |

---

\(a\) Entries for first 18 programs not available because times were not recorded.

\(b\) Includes time for typing in program and punching out paper tape.
### B.3 Table of check program usage (Part 1)

<table>
<thead>
<tr>
<th>Program Number</th>
<th>Session Number</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1-2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6-8a</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>S  T  C</td>
<td>S  T  C</td>
<td>S  T  C</td>
<td>S  T  C</td>
<td>S  T  C</td>
<td>S  T  C</td>
</tr>
<tr>
<td>1</td>
<td>7 18 4</td>
<td>1 2 1</td>
<td>0 1 1</td>
<td>1 1 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>9 22 6</td>
<td></td>
<td></td>
<td>1 3 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>9 11 8</td>
<td></td>
<td>1 4 0</td>
<td>1 2 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4 12 2</td>
<td></td>
<td>3 6 1</td>
<td>2 9 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1 1 0 1 3 1</td>
<td>4 6 3</td>
<td>1 3 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1 2 1 1 3 0</td>
<td>3 7 3</td>
<td>1 3 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>2 5 2</td>
<td></td>
<td>3 4 2</td>
<td>1 2 1</td>
<td>5 7 5</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1 1 0</td>
<td></td>
<td>1 1 0</td>
<td>4 1 3</td>
<td>0 4 1 0 3 1 5 0</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>1 5 0</td>
<td>2 9 0</td>
<td>2 5 0</td>
<td>4 6 0</td>
<td>1 1 0</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>2 5 0</td>
<td>1 3 0</td>
<td>2 3 1</td>
<td>3 9 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>2 5 1</td>
<td></td>
<td></td>
<td>1 1 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td>1 4 1</td>
<td>1 4 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td>1 1 1</td>
<td>1 1 1</td>
<td>5 6 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td>2 3 1</td>
<td>2 3 2</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
<td></td>
<td>1 2 1</td>
<td>1 4 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
<td></td>
<td>1 1 0</td>
<td></td>
<td>1 1 1</td>
<td>1 3 1</td>
</tr>
<tr>
<td>Total</td>
<td>34 72 25</td>
<td>4 13 2</td>
<td>23 46 11</td>
<td>14 39 4</td>
<td>30 57 23</td>
<td>9 25 4</td>
</tr>
</tbody>
</table>

**Students Programming this session**

- **9**
- **2**
- **9**
- **10**
- **10**
- **6**

- *a* Sessions combined because teacher report was not run every session.
- *b* S - Number of students trying this program
- T - Number of total tries of this program
- C - Number of students completing this program
### B.3 Table of check program usage (Part 2)

<table>
<thead>
<tr>
<th>Program Number</th>
<th>Session Number</th>
<th>10</th>
<th>11-12&lt;sup&gt;b&lt;/sup&gt;</th>
<th>13</th>
<th>14-16&lt;sup&gt;b&lt;/sup&gt;</th>
<th>17-19</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>S&lt;sup&gt;c&lt;/sup&gt;</td>
<td>T&lt;sup&gt;c&lt;/sup&gt;</td>
<td>C</td>
<td>S&lt;sup&gt;c&lt;/sup&gt;</td>
<td>T&lt;sup&gt;c&lt;/sup&gt;</td>
<td>C</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>6</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>5</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>5</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>16</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>17</td>
<td>1</td>
<td>5</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>1</td>
<td>5</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>19</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>20</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>21</td>
<td>2</td>
<td>4</td>
<td>0</td>
<td>2</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>22</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>23</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>24</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>25</td>
<td>3</td>
<td>7</td>
<td>2</td>
<td>3</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>26</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>27</td>
<td>2</td>
<td>9</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>28</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>9</td>
<td>3</td>
</tr>
<tr>
<td>29</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>30</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>31</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

**Total**       | 13             | 23 | 7                | 13   | 19               | 3    | 13   | 26   | 7    | 34   | 80  | 27   | 37   | 112  | 30   |

**Students Programming this session**

| Programming this session | 8 | 5 | 3 | 6 | 5 |

---

<sup>a</sup> Program numbers left out were not tried during these sessions.

<sup>b</sup> Sessions combined because teacher report was not run every session

<sup>c</sup> S - Number of students trying this program

T - Number of total tries of this program

C - Number of students completing this program
### B.3 Table of check program usage (Part 3)

<table>
<thead>
<tr>
<th>Program Number</th>
<th>Students Successful</th>
<th>Students Unsuccessful</th>
<th>Tries S Students *</th>
<th>Tries U Students *</th>
<th>Average Tries S Students *</th>
<th>Average Tries U Students *</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7</td>
<td>1</td>
<td>8</td>
<td>22</td>
<td>3</td>
<td>25</td>
</tr>
<tr>
<td>2</td>
<td>9</td>
<td>2</td>
<td>11</td>
<td>25</td>
<td>4</td>
<td>29</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>1</td>
<td>11</td>
<td>19</td>
<td>1</td>
<td>20</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>2</td>
<td>8</td>
<td>20</td>
<td>8</td>
<td>28</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>1</td>
<td>6</td>
<td>13</td>
<td>1</td>
<td>14</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>0</td>
<td>6</td>
<td>18</td>
<td>0</td>
<td>18</td>
</tr>
<tr>
<td>7</td>
<td>12</td>
<td>0</td>
<td>12</td>
<td>21</td>
<td>0</td>
<td>21</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>3</td>
<td>7</td>
<td>22</td>
<td>13</td>
<td>35</td>
</tr>
<tr>
<td>9</td>
<td>4</td>
<td>4</td>
<td>8</td>
<td>15</td>
<td>15</td>
<td>30</td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>4</td>
<td>7</td>
<td>16</td>
<td>18</td>
<td>34</td>
</tr>
<tr>
<td>11</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>12</td>
<td>0</td>
<td>12</td>
</tr>
<tr>
<td>12</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>8</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>13</td>
<td>9</td>
<td>0</td>
<td>11</td>
<td>12</td>
<td>0</td>
<td>12</td>
</tr>
<tr>
<td>14</td>
<td>5</td>
<td>1</td>
<td>6</td>
<td>7</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>15</td>
<td>6</td>
<td>0</td>
<td>6</td>
<td>16</td>
<td>0</td>
<td>16</td>
</tr>
<tr>
<td>16</td>
<td>6</td>
<td>1</td>
<td>7</td>
<td>21</td>
<td>7</td>
<td>28</td>
</tr>
<tr>
<td>17</td>
<td>4</td>
<td>1</td>
<td>5</td>
<td>15</td>
<td>7</td>
<td>22</td>
</tr>
<tr>
<td>18</td>
<td>6</td>
<td>0</td>
<td>6</td>
<td>21</td>
<td>0</td>
<td>21</td>
</tr>
<tr>
<td>19</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>14</td>
<td>0</td>
<td>14</td>
</tr>
<tr>
<td>20</td>
<td>3</td>
<td>2</td>
<td>5</td>
<td>8</td>
<td>4</td>
<td>12</td>
</tr>
<tr>
<td>21</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>9</td>
<td>0</td>
<td>9</td>
</tr>
<tr>
<td>22</td>
<td>5</td>
<td>0</td>
<td>5</td>
<td>15</td>
<td>0</td>
<td>15</td>
</tr>
<tr>
<td>23</td>
<td>4</td>
<td>1</td>
<td>5</td>
<td>23</td>
<td>4</td>
<td>27</td>
</tr>
<tr>
<td>24</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>25</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>7</td>
<td>5</td>
<td>12</td>
</tr>
<tr>
<td>26</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>27</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>28</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>9</td>
<td>0</td>
<td>9</td>
</tr>
<tr>
<td>29</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>30</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>31</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>10</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>32</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>33</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>7</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>34</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>35</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>36</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>37</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>38</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

| Average for all Programs | 3.8 | .8 | 4.6 | 10.9 | 2.6 | 13.5 | 2.9 | 3.4 | 3.0 |

a S - Successful       U - Unsuccessful
**Sample Check Program**

1  COM L[1],Z[1]

20  DEF FNZ(X)=INT(RND(0)*X)+1

30  LET P9=2

40  PRINT "PROG";P9

100  PRINT "STUDENT NO.";

110  INPUT L9

115  IF L9<1 OR L9>18 THEN 110

120  CALL (5,L9,P9,T9)

124  LET L[1]=L9

126  LET Z[1]=P9

130  IF T9<7 THEN 485

140  GOTO 9000

150  STOP

485  CALL (4,L9,P9,T9+1)

490  LET I9=1

500  LET X1=FNZ(100)-50

510  LET X2=FNZ(100)-25

520  LET D9=X2-X1

530  IF D9<0 THEN 550

540  LET D9=-D9

550  LET X3=FNZ(100)-75

560  LET E9=X3-X1

570  IF E9<0 THEN 590

580  LET E9=-E9

590  IF D9<E9 THEN 610

600  LET D9=E9

610  LET E9=X3-X2

620  IF E9<0 THEN 640

630  LET E9=-E9

640  IF D9<E9 THEN 660

650  LET D9=E9

660  LET E9=-D9

690  REM

9000  IF D#D9 OR E#E9 THEN 9500

9200  PRINT "OK";I9

9205  LET I9=I9+1

9210  IF I9<6 THEN 500

9215  IF L9<1 OR L9>18 OR T9<0 OR T9>7 THEN 9260

9220  PRINT "PROG";P9;"CORRECT"

9240  CALL (4,L9,2*T9+9)

9250  STOP

9260  PRINT "USING VARIABLE NAME ENDING IN 9"

9270  STOP

9500  PRINT "PROG";P9;"IN ERROR"

9600  PRINT "X1=";X1

9610  PRINT "X2=";X2

9620  PRINT "X3=";X3

9680  PRINT "","YOUR ANSWER","CORRECT ANSWER"

9690  PRINT TAB(10);"D",D9

9700  PRINT TAB(10);"E",E9

9740  STOP

9999  END
B.5 Sample correct solutions

PLIST 1400

1400 LET N2=0
1410 FOR I=1 TO N1
1420 FOR J=1 TO N1
1430 LET N2=N2+1
1440 LET B[N2]=A[I]
1450 LET C[N2]=A[J]
1460 NEXT J
1470 NEXT I
1480 GOTO 9000
9000 IF N2=N9 THEN 9500
9010 STOP

PLIST 1800

1800 FOR I=1 TO M1
1810 FOR J=1 TO N2
1830 NEXT J
1840 GOTO 1980
1850 NEXT I
1900 FOR I=1 TO N2
1910 FOR J=I+1 TO N2
1940 NEXT J
1950 NEXT I
1960 LET F=1
1970 GOTO 9000
1980 LET F=0
1990 GOTO 9000
9000 STOP

PLIST 2200

2200 LET X=ABS(X2-X1)
2210 LET Y=ABS(Y2-Y1)
2220 LET D=SQRT(X2+Y2)
2230 GOTO 9000
900 STOP

Correct Solution
HW Problem
No. 3

Correct Solution
HW Problem
No. 5

Correct Solution
HW Problem
No. 7
<table>
<thead>
<tr>
<th>Session No.</th>
<th>Number of times TPI program tried</th>
<th>Total per Session</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>4(7)b</td>
<td>3(11)</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6-8a</td>
<td>1(1)</td>
<td>1(3)</td>
</tr>
<tr>
<td>9</td>
<td>1(0)</td>
<td>1(2)</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11-12a</td>
<td>1(1)</td>
<td>1(0)</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>14-16a</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>17-19a</td>
<td>0</td>
<td>1(2)</td>
</tr>
<tr>
<td>Total</td>
<td>7(9)</td>
<td>10(27)</td>
</tr>
<tr>
<td>Average</td>
<td>1.3</td>
<td>2.7</td>
</tr>
</tbody>
</table>

### Writing time

<table>
<thead>
<tr>
<th></th>
<th>115d</th>
<th>75</th>
<th>75</th>
<th>115</th>
<th>105</th>
<th>85</th>
<th>55</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total</td>
<td>470</td>
<td>255</td>
<td>163</td>
<td>143</td>
<td>168</td>
<td>108</td>
<td>163</td>
</tr>
</tbody>
</table>

### Debugging time

<table>
<thead>
<tr>
<th></th>
<th>89e</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total</td>
<td>179e</td>
</tr>
</tbody>
</table>

*a These sessions are combined because teacher report was not run every session.
*b Number in parentheses represents number of wrong responses.
*c Entries for first three TPI programs not available because times were not recorded.
*d Time in minutes.
*e Average instead of total.
*f Includes time for typing in program and punching out paper tape.
A CONIC SECTN WHERE EQUATION IS \( A*X^2 + C*Y^2 + D*X + E*Y + F = 0 \) IS
A HYPERBOLA IF
1. \( A \neq C \)
2. \( A \neq 0 \)
3. \( C \neq 0 \)
4. \( A \neq C \neq 0 \)
5. \( A \neq C \neq 0 \)

A ELLIPSE IF
73
A PARABOLA IF
76
A CIRCLE IF
71
CORRECT NO. 1.

TO ELIMINATE THE X*Y TERM IN \( A*X^2 + 3*X*Y + C*Y^2 + D*X + E*Y + F = 0 \) TAKE THE ANGLE WHERE
\( \tan(2*\theta) = B/A - C \) IF \( A \neq C \).

IF \( \tan(2*\theta) = 3 \neq 0 \), \( *X*Y \) 0, ESTIMATE THE ANGL (IN DEG) THE AXES MUST BE ROTATED TO ELIMINATE THE X*Y TERM
73
T=35
ANS=35.7825 -35 IS CLOSE ENOUGH NO. 1

IF \( A \neq C \) BEFORE A ROTATION, WILL \( A \neq C \) AFTER THE ROTATION \( (Y=1, X=0) \).
CORRECT! NO. 1

IF \( A \neq C \) BEFORE A TRANSLATION, WILL \( A \neq C \) AFTER THE TRANSLATION \( T \).
CORRECT! NO. 1

TO CONVERT FROM RECT COORDINATES \((X, Y)\) TO POLAR COORD \((R, \theta)\) LET \( R = \sqrt{X^2 + Y^2} \) & \( \theta = \tan^{-1}(Y/X) \) WHERE \( \theta \) IS IN RADS
ESTIMATE POLAR COORD OF \((4, 0)\)
74.0
R = 4 & \( \theta = 0 \) RADIANS
CLOSE ENOUGH

TO CONVERT FROM RAD TO DEG USE \( 1 \text{ RAD} = 57.3 \text{ DEG} \).
T=PI, LET \( \pi \times 4 \times 4.5 \times 1/4 \) SINCE \( 45 \text{ DEG} = \pi/4 \) RADS
-5.928
-28
ANS=28.6479 DEG-28 IS CLOSE ENOUGH NO. 1

3 WRONG FOR THY 1 OF CAL # 8
READY
**D.1 Letter to Jeff Spain**

**STORAGE AREAS**

**QUIZ** is an array of 90 words stored as follows:

<table>
<thead>
<tr>
<th>Bit</th>
<th>Word 15</th>
<th>Word 12</th>
<th>Word 11</th>
<th>Word 8</th>
<th>Word 7</th>
<th>Word 4</th>
<th>Word 3</th>
<th>Word 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Student 1</td>
<td>Student 1</td>
<td>Student 1</td>
<td>Student 1</td>
<td>Student 1</td>
<td>Student 1</td>
<td>Student 1</td>
<td>Student 1</td>
</tr>
<tr>
<td>2</td>
<td>Correct Quiz 1</td>
<td>Correct Quiz 1</td>
<td>Correct Quiz 1</td>
<td>Correct Quiz 1</td>
<td>Correct Quiz 1</td>
<td>Correct Quiz 1</td>
<td>Correct Quiz 1</td>
<td>Correct Quiz 1</td>
</tr>
<tr>
<td>3</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
</tr>
<tr>
<td>4</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
</tr>
<tr>
<td>5</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
</tr>
<tr>
<td>6</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
</tr>
<tr>
<td>7</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
</tr>
<tr>
<td>8</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
</tr>
<tr>
<td>9</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
</tr>
<tr>
<td>10</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
</tr>
<tr>
<td>11</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
</tr>
<tr>
<td>12</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
</tr>
<tr>
<td>13</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
</tr>
<tr>
<td>14</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
</tr>
<tr>
<td>15</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
</tr>
</tbody>
</table>

**PROG** is an array of 180 words stored as follows:

<table>
<thead>
<tr>
<th>Bit</th>
<th>Word 15</th>
<th>Word 12</th>
<th>Word 11</th>
<th>Word 8</th>
<th>Word 7</th>
<th>Word 4</th>
<th>Word 3</th>
<th>Word 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Student 1</td>
<td>Student 1</td>
<td>Student 1</td>
<td>Student 1</td>
<td>Student 1</td>
<td>Student 1</td>
<td>Student 1</td>
<td>Student 1</td>
</tr>
<tr>
<td>2</td>
<td>Times Tried Prog 1</td>
<td>Times Tried Prog 1</td>
<td>Times Tried Prog 1</td>
<td>Times Tried Prog 1</td>
<td>Times Tried Prog 1</td>
<td>Times Tried Prog 1</td>
<td>Times Tried Prog 1</td>
<td>Times Tried Prog 1</td>
</tr>
<tr>
<td>3</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
</tr>
<tr>
<td>4</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
</tr>
<tr>
<td>5</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
</tr>
<tr>
<td>6</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
</tr>
<tr>
<td>7</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
</tr>
<tr>
<td>8</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
</tr>
<tr>
<td>9</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
</tr>
<tr>
<td>10</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
</tr>
<tr>
<td>11</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
</tr>
<tr>
<td>12</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
</tr>
<tr>
<td>13</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
</tr>
<tr>
<td>14</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
</tr>
<tr>
<td>15</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
</tr>
</tbody>
</table>

**CAI** is an array of 90 words stored as follows:

<table>
<thead>
<tr>
<th>Bit</th>
<th>Word 15</th>
<th>Word 14</th>
<th>Word 13</th>
<th>Word 8</th>
<th>Word 7</th>
<th>Word 6</th>
<th>Word 5</th>
<th>Word 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Student 1</td>
<td>Student 1</td>
<td>Student 1</td>
<td>Student 1</td>
<td>Student 1</td>
<td>Student 1</td>
<td>Student 1</td>
<td>Student 1</td>
</tr>
<tr>
<td>2</td>
<td>Times Cat 1</td>
<td>Times Cat 1</td>
<td>Times Cat 1</td>
<td>Times Cat 1</td>
<td>Times Cat 1</td>
<td>Times Cat 1</td>
<td>Times Cat 1</td>
<td>Times Cat 1</td>
</tr>
<tr>
<td>3</td>
<td>Entries Range from 0 to 43</td>
<td>Entries Range from 0 to 43</td>
<td>Entries Range from 0 to 43</td>
<td>Entries Range from 0 to 43</td>
<td>Entries Range from 0 to 43</td>
<td>Entries Range from 0 to 43</td>
<td>Entries Range from 0 to 43</td>
<td>Entries Range from 0 to 43</td>
</tr>
<tr>
<td>4</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
</tr>
<tr>
<td>5</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
</tr>
<tr>
<td>6</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
</tr>
<tr>
<td>7</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
</tr>
<tr>
<td>8</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
</tr>
<tr>
<td>9</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
</tr>
<tr>
<td>10</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
</tr>
<tr>
<td>11</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
</tr>
<tr>
<td>12</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
</tr>
<tr>
<td>13</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
</tr>
<tr>
<td>14</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
</tr>
<tr>
<td>15</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
</tr>
</tbody>
</table>

---
SUBROUTINE 2 ; CALL [2, A(2), S(1)]

DESCRIPTION: Take data stored in S array (dimensioned 15) and store in appropriate 4-bit position of QUIZ array as determined by A(2). The S array contains each student's score on quiz A(2).

```
<table>
<thead>
<tr>
<th>A(2)</th>
<th>K</th>
<th>W</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>17</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>18</td>
<td>3</td>
<td>5</td>
</tr>
</tbody>
</table>
```

Note: A(2) in Main Program should not be changed.
SUBROUTINE 3; CALL [3, L, J, SCORE]

DESCRIPTION: RETRIEVE SCORE OF STUDENT L ON QUIZ J

LOAD QUIZ [5*(L-1)+W] INTO REGISTER A
SHIFT REGISTER A 'K' 4-BIT GROUPS TO THE LEFT
'AND' REGISTER A WITH 170000B
STORE REGISTER A INTO SCORE
RETURN

CHECK DATA

<table>
<thead>
<tr>
<th>J</th>
<th>K</th>
<th>W</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>17</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>18</td>
<td>1</td>
<td>5</td>
</tr>
</tbody>
</table>
SUBROUTINE 4;  CALL [4, L, J, N]

DESCRIPTION: Store N, the number of times student L tried to program J, in appropriate 4-bit position of PROG array.

FLOWCHART:

- W = 1
- IF J > 4 THEN
  - W = W + 1
  - J = J - 4
- K = J - 1
- LOAD PROG[10*(L-1) + W] INTO REGISTER A
- SHIFT REGISTER A 'K' 4-BIT GROUPS TO THE LEFT
- 'AND' REGISTER A WITH 007777
- 'OR' REGISTER A WITH N
- SHIFT REGISTER A '4-K' 4-BIT GROUPS TO THE LEFT
- STORE REGISTER A INTO PROG[10*(L-1) + W]
- RETURN

NEED TO ERASE OLD CONTENTS OF PROG
SUBROUTINE 5; CALL [5, L, J, N]

DESCRIPTION: RETRIEVE N, THE NUMBER OF TIMES STUDENT L TRIED PROGRAM J.

```
W = 1

J > 4

W = W + 1
J = J - 4

K = J - 1

LOAD PROG [10*(L-1) + W] INTO REGISTER A

SHIFT REGISTER A 'K' 4-BIT GROUPS TO THE LEFT

'AND' REGISTER A WITH 170000B

STORE REGISTER A INTO N

RETURN
```

ONLY DIFFERENCE FROM SUB3
SUBROUTINE 6; CALL [G, L, J, N1, N2]


```
W=1

J > 2?

YES

W = W + 1

J = J - 2

NO

K = 2**((J-1))

LOAD N2 INTO REG B

SHIFT REG B 2 BITS TO THE RIGHT

'OR' REGISTER B WITH N1

LOAD CAI [5* (L-1) + W] INTO REGISTER A

SHIFT REGISTER A 'K' 4-BIT GROUPS TO THE LEFT

'AND' REGISTER A WITH 000377B

'OR' REGISTER A WITH REGISTER B

SHIFT REGISTER A 'A-K' 4-BIT GROUPS TO THE LEFT

STORE REGISTER A INTO CAI [5* (L-1) + W]

RETURN
```

CHECK DATA

<table>
<thead>
<tr>
<th>J</th>
<th>K</th>
<th>W</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>9</td>
<td>10</td>
</tr>
</tbody>
</table>
SUBROUTINE 7; CALL [7, L, J, N1, N2]

DESCRIPTION: Retrieve N1, the number of times student L tried CAI J, and N2, the number of incorrect answers the last time student L tried CAI J.

```
W = 1

IF J < 2 THEN
    W = W + 1
    J = J - 2
ELSE
    K = 2 * (J - 1)

LOAD CAI [5 * (L - 1) + W] INTO REGISTER A
SHIFT REGISTER A 'K' 4-BIT GROUPS TO THE LEFT
'AND' REGISTER A WITH 110000B
STORE REGISTER A INTO N1

LOAD CAI [5 * (L - 1) + W] INTO REGISTER A
SHIFT REGISTER A 'K' 4-BIT GROUPS TO THE LEFT
'AND' REGISTER A WITH 037400B
SHIFT REG A 2 BITS TO THE LEFT
STORE REGISTER A INTO N2
RETURN
```
The following program can be used for checking out the subroutines:

```
5 DIM $18
10 FOR $=1 TO 18
20 FOR L=1 TO 18
30 LET $=INT(L/3)
40 NEXT L
50 CALL [2, J, $]
60 NEXT J
100 FOR L=1 TO 18
110 FOR J=1 TO 38
120 LET N=INT(J/3)
130 CALL [4, L, J, N]
140 NEXT J
150 FOR J=1 TO 10
160 LET N1=INT(J/3)
170 LET N2=J
180 CALL [6, L, J, N1, N2]
185 NEXT J
190 NEXT L
200 FOR L=1 TO 18
210 FOR J=1 TO 18
220 CALL [3, L, J, $]
230 PRINT $;
240 NEXT J
250 PRINT
260 NEXT L
```
300 FOR L=1 TO 18
310 FOR J=1 TO 38
320 CALL [5, L, J, N]
330 PRINT N;
340 NEXT J
350 PRINT
360 NEXT L

400 FOR L=1 TO 18
410 FOR J=1 TO 10
420 CALL [7, L, J, N1, N2]
430 PRINT N1; N2
440 NEXT J
450 PRINT
460 NEXT L

500 END
D.2 Flowcharts of reporting programs

Teacher Report
Program A

ZERO ALL COUNTERS

READ IN QUIZ KEY

FIRST CARD 'O'?

PRINT "FIRST CARD NOT 'O' CARD"

STOP

READ IN STUDENT ANSWERS

STUDENT ID = 99?

STUDENT ID > 19 OR < 19?

STUDENT QUIZ NUMBER CORRECT?

ADD 1 TO NUMBER OF STUDENTS TAKING QUIZ

ADD STUDENT HW TIME TO HW TIME TOTAL

I = 1

STUDENT ANSWER TO QUESTION I = KEY?

ADD 1 TO NUMBER CORRECT FOR THIS STUDENT

ADD 1 TO NUMBER CORRECT FOR THIS QUESTION

I = I + 1

YES

NO

STUDENT QUIZ NUMBER CORRECT?

NO
CALL SUBROUTINE 2 TO STORE QUIZ RESULTS
PRINT NUMBER OF STUDENTS MISSENG EACH QUIZ QUESTION
PRINT HEADING FOR STUDENT CUMULATIVE QUIZ RECORD

L = 1
J = 1

CALL SUBROUTINE 3 TO GET RESULTS FOR STUDENT L ON QUIZ J
PRINT RESULTS OF STUDENT L ON QUIZ J

J = J + 1

IF J'S CURRENT QUIZ IS YES THEN

L = L + 1

IF L ≤ 18

PRINT AVERAGE TIME SPENT ON HOMEWORK THIS SESSION
STUDENT NUMBER 'L' AND PROBLEM NUMBER 'P' SAVED FROM CHECK PROGRAM

PRINT "WOULD YOU LIKE YOUR QUIZ SCORE?"

INPUT Q

Q = 0

YES

1

NO

PRINT "FOR QUIZ NO."

INPUT J9

J9 > 18

YES

CALL SUB 3 TO GET SCORE 'S' OF STUDENT 'L' ON QUIZ 'J9'

NO

PRINT "SCORE IS"; S

S > 5

YES

1

NO

CALL SUB 7 TO GET NUMBER OF TRIES 'N1', AND NUMBER WROTE 'N2', FOR STUDENT 'L' ON QUIZ 'J9'

N1 > 0

YES

NO

TAKE TPF ASSOCIATED WITH QUIZ
2

PRINT "CALC NO.
"TRIED", "WRONG"

J = 1

CALL SUB7 TO GET NUMBER
OF TRIES, N1, AND NUMBER WRONG
N2, FOR STUDENT 'L' ON TPI 'J'

N1 = 0

NO → PRINT J, N1, N2

J = J + 1

NO → J ≤ 10

CALL SUB5 TO GET NUMBER
OF TIMES 'T' STUDENT 'L'
TRIED PROGRAM P9

T < 8

NO → J = P9 + 1

CALL SUB5 TO GET NUMBER
OF TIMES 'T' STUDENT 'L'
TRIED PROGRAM 'J'

T < 8

YES ← TRY PROG J

NO

J = J + 1

YES

J < 38

NO

STOP

CALL SUB7 TO GET NUMBER
OF TIMES N1 AND NUMBER
WRONG N2 FOR STUDENT 'L' ON
TPI ASSOCIATED WITH PROGRAM P9

N1 > 0

NO

N2 < 5

YES OR N1 ≤ 3

NO

YES

RETRY PROG P9

TAKE TPI ASSOCIATED WITH P9
1. Given set \( A = \{ 0, 2, -5, 0, (7, 12), 6, 9, 100 \} \) and set \( B = \{ a, b, c, 5, 0, (12, 7), -6, 100, 9 \} \), find \( A \cap B \) and \( A \cup B \).

2. Write the converse of "Basic programmers are not very smart."

3. Graph the solution set of \( x > 7 \) and \( x < 9 \).

4. Define the four quadrants and give an example of a point in each one.

5. Is \( \{ (0, 7), (1, 8), (2, 9), (3, 4) \} \) a function? Why or why not? What is its range and domain?

6. How long is the segment connecting \((-1, 8)\) and \((3, -2)\)? What is its midpoint? Its equation? Its slope?

7. What is the area of a triangle whose vertices are \((-1, 8), (2, 6), (-3, 7)\)?

8. What is the equation of the line perpendicular to \( y = 3x + 6 \) and containing \((0, 0)\)?

9. What is the intersection of \( y = 3x + 6 \) and \( y = -3x + 2 \)?

10. The center and radius of the circle \( x^2 + y^2 - 6y - 12 \) are?
11. In the following flow chart the values of A, B, C on input are respectively 2, 1, 0. What are the values of A, B, C, age, & I on output?

(Do not spend more than 5 minutes on this problem.)

```
INPUT A, B, C

AGE = 2
I = 0

A > B

AGE = AGE + B
I = I + 1

A = C + I

I > B

AGE = AGE - C

PRINT A, B, C, AGE, I
```
12. Find the errors in the following BASIC program

```
START
LET X = 12
INPUT A, BC, DX2
FOR I = 1 TO X*2 STEP C
D(I) = -A + BC + 2D
IF D(I) > = C THEN 20
I = I + A
20 REM SKIP I
NEXT C
STOP
```

13. If \( A = 5 \), \( B = 6 \), \( C = 12 \), \( D = 0 \), and \( E = -1 \), then the statement \( F = A + B/C + D* (A + E) \) places what value in \( F \)?

14. Write a program to generate 10 random numbers, list these numbers and also their minimum, maximum, and mean.

15. Write a program to determine the equation of a line given the coordinates of two points on the line as input.


17. Draw a flowchart of problem 15.
## PRETEST SUMMARY

### Part 1 - Analytic Geometry

<table>
<thead>
<tr>
<th>Student ID</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>61</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>10</td>
<td>5</td>
<td>10</td>
<td>61</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>80</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>80</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>10</td>
<td>10</td>
<td>0</td>
<td>60</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>10</td>
<td>3</td>
<td>5</td>
<td>0</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>10</td>
<td>63</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>0</td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>10</td>
<td>53</td>
</tr>
<tr>
<td>7</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>73</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>5</td>
<td>10</td>
<td>44</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>5</td>
<td>53</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>10</td>
<td>7</td>
<td>5</td>
<td>5</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>55</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>10</td>
<td>0</td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>0</td>
<td>10</td>
<td>10</td>
<td>65</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>7</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>59</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>5</td>
<td>48</td>
</tr>
<tr>
<td>15</td>
<td>3</td>
<td>0</td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>31</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>10</td>
<td>0</td>
<td>10</td>
<td>5</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>9</td>
<td>10</td>
<td>52</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>9</td>
<td>0</td>
<td>10</td>
<td>5</td>
<td>10</td>
<td>3</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>77</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>10</td>
<td>0</td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>7</td>
<td>0</td>
<td>10</td>
<td>10</td>
<td>67</td>
<td></td>
</tr>
</tbody>
</table>

### Part 2 - BASIC

<table>
<thead>
<tr>
<th>Student ID</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>7</td>
<td>10</td>
<td>13</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>50</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>8</td>
<td>10</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>32</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>6</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>16</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>39</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>8</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>18</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>5</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>15</td>
</tr>
<tr>
<td>8</td>
<td>20</td>
<td>2</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>32</td>
</tr>
<tr>
<td>9</td>
<td>8</td>
<td>5</td>
<td>10</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>31</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>2</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>12</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>9</td>
</tr>
<tr>
<td>12</td>
<td>8</td>
<td>4</td>
<td>5</td>
<td>16</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>36</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>4</td>
<td>10</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>17</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>6</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>16</td>
</tr>
<tr>
<td>15</td>
<td>3</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>13</td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>4</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>14</td>
</tr>
<tr>
<td>17</td>
<td>20</td>
<td>9</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>39</td>
</tr>
<tr>
<td>18</td>
<td>20</td>
<td>3</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>33</td>
</tr>
</tbody>
</table>

### Possible Score

<table>
<thead>
<tr>
<th>Test</th>
<th>10</th>
<th>10</th>
<th>10</th>
<th>10</th>
<th>10</th>
<th>10</th>
<th>10</th>
<th>100</th>
</tr>
</thead>
</table>

### Class Average

<table>
<thead>
<tr>
<th>Part</th>
<th>8</th>
<th>7</th>
<th>8</th>
<th>5</th>
<th>3</th>
<th>7</th>
<th>1</th>
<th>8</th>
<th>6</th>
<th>8</th>
</tr>
</thead>
</table>

**Related Sections of Text**

- 1.1
- 1.2
- 1.5
- 1.7
- 1.8
- 2.2
- 2.4
- 3.1
- 3.2
- 3.1
- 3.2
- 3.2

**Score** 24
THE LINE, Test 4

I. Find those values of \( x \) for which \( 1 \leq |5 - 4x| \) and \(-1 < x < \frac{7}{4}\).

1. Which of the following is the correct answer to problem I above?
   (A) \(-1 < x < 7\)
   (B) \(1 \leq x \leq \frac{7}{4}\)
   (C) \(-1 < x \leq 1\)
   (D) \(\frac{7}{4} \leq x < \frac{7}{2}\)
   (E) those \( x \) in the intersection of \(-1 < x \leq 1\) and \(\frac{7}{4} \leq x < \frac{7}{2}\)
   (F) no values of \( x \)

II. Find the distance and the directed distance (if it exists) from the point \( P_1 \) to the point \( P_2 \) whose coordinates are given below.

(i) \( P_1 (8, 0), P_2 (3, 0) \)

2. The directed distance is
   (A) \(-11\)
   (B) \(-5\)
   (C) \(5\)
   (D) \(11\)
   (E) there is none

3. The distance is
   (A) \(-11\)
   (B) \(-5\)
   (C) \(5\)
   (D) \(1\)
   (E) there is none

(ii) \( P_1 (0, -Q), P_2 (0, d) \)

4. The directed distance is
   (A) \(d - Q\)
   (B) \(d + Q\)
   (C) \(Q - d\)
   (D) \(-Q - d\)
   (E) there is none

5. The distance is
   (A) \(|d - Q|\)
   (B) \(|d + Q|\)
   (C) \(|Q - d|\)
   (D) \(|-Q - d|\)
   (E) \(|d - Q|\)
   (F) \(|d + Q|\)
   (G) \(|Q - d|\)
   (H) \(|-Q - d|\)

(iii) \( P_1 (7, -5), P_2 (1, 2) \)

6. The directed distance is
   (A) \(-\sqrt{55}\)
   (B) \(-\sqrt{73}\)
   (C) \(-\sqrt{45}\)
   (D) \(\sqrt{13}\)
   (E) \(\sqrt{3}\)
   (F) \(\sqrt{75}\)
   (G) \(\sqrt{5}\)
   (H) there is none

7. The distance is
   (A) \(-\sqrt{55}\)
   (B) \(-\sqrt{73}\)
   (C) \(-\sqrt{45}\)
   (D) \(\sqrt{13}\)
   (E) \(\sqrt{3}\)
   (F) \(\sqrt{75}\)
   (G) \(\sqrt{5}\)
   (H) there is none
III. In these problems use your common sense to assign the appropriate slope to each line.

8. The slope is
(A) -6
(B) -2
(C) -\(\frac{1}{6}\)
(D) -\(\frac{1}{2}\)
(E) \(\frac{1}{2}\)
(F) 2
(G) 6

9. The slope is
(A) -6
(B) -2
(C) -\(\frac{1}{6}\)
(D) -\(\frac{1}{2}\)
(E) \(\frac{1}{2}\)
(F) \(\frac{1}{3}\)
(G) 2
(H) 6

IV. Given the lines \(L_1\) through \((-2, 5)\) and \((6, 8)\) and \(L_2\) through \((5, -2)\) and \((-8, -7)\), are they parallel, perpendicular or neither?

10. In the problem above the lines were
(A) parallel
(B) perpendicular
(C) neither

V. Find the equation of the line that passes through the points \((1, 5)\) and \((2, 1)\). Put your answer in the form \(y = Ax + B\) and answer the question below.

11. The sum of the coefficients \(A + B\) is
(A) -7
(B) -5
(C) -3
(D) -\(\frac{1}{2}\)
(E) 1
(F) \(\frac{1}{2}\)
(G) 5
(H) 7\(\frac{2}{3}\)

VI. Find the equation of the line through the point \((-8, 4)\) parallel to the line \(2x - 8y + 9 = 0\). Put your answer in the form \(y = Ax + B\) and answer the question below.
12. The sum of the coefficients $A + B$ is
(A) $-24$
(B) $-5$
(C) $\frac{11}{2}$
(D) $1\frac{3}{4}$
(E) $6\frac{1}{2}$
(F) $24$
(G) $40$
(H) $24$

VII. Find the intercepts and asymptotes, and test for symmetry of the curve whose equation is $y(x^2 - 4) = 8x^2$.

13. The $x$-intercept(s) are:
(A) $-8$
(B) $-\sqrt{8}, +\sqrt{8}$
(C) $-\sqrt{8}, 0, +\sqrt{8}$
(D) $-2, +2$
(E) $-2, 0, +2$
(F) $0$
(G) $8$
(H) There are none

14. The $y$-intercept(s) are:
(A) $-8$
(B) $-\sqrt{8}, +\sqrt{8}$
(C) $-\sqrt{8}, 0, +\sqrt{8}$
(D) $-2, +2$
(E) $-2, 0, +2$
(F) $0$
(G) $8$
(H) There are none

15. The horizontal asymptote(s) pass through
(A) $(0, -8)$
(B) $(0, -8)$ and $(0, +8)$
(C) $(0, -2)$
(D) $(0, -2)$ and $(0, +2)$
(E) $(0, 0)$
(F) $(0, 2)$
(G) $(0, 8)$
(H) There are none

16. The vertical asymptote(s) pass through
(A) $(-8, 0)$
(B) $(-8, 0)$ and $(+8, 0)$
(C) $(-2, 0)$
(D) $(-2, 0)$ and $(+2, 0)$
(E) $(0, 0)$
(F) $(+2, 0)$
(G) $(+8, 0)$
(H) There are none

17. The curve is symmetric with respect to:
(A) $x$-axis only
(B) $y$-axis only
(C) origin only
(D) $x$ and $y$ axes only
(E) $x$-axis and origin
(F) $y$-axis and origin
(G) $x$ and $y$ axes and origin
(H) None of these
VIII. Sketch the curve which has the following properties:
1. It passes through the points \((4, \frac{3}{2})\) and \((-2, -1)\).
2. It has no x-intercept.
3. The y-intercept is \(-\frac{4}{3}\).
4. It is symmetric with respect to the y-axis only.
5. It has a horizontal asymptote which passes through \((0, 0)\).
6. It has 2 vertical asymptotes which pass through \((-3, 0)\) and \((3, 0)\).
THE CONICS, Test A

1. Write the equation $5x^2 + 5y^2 + 30x - 10y + 14 = 0$ in standard form and identify the conic. Then answer the questions that follow.

1. The conic is
   (A) a circle
   (B) a parabola
   (C) an ellipse
   (D) a hyperbola
   (E) degenerate

2. The focus is (or foci are) at
   (A) $(1, -3)$
   (B) $(-3, 1)$
   (C) $(1 - \sqrt{5}, -3)$ and $(1 + \sqrt{5}, -3)$
   (D) $(-3 - \sqrt{5}, 1)$ and $(-3 + \sqrt{5}, 1)$
   (E) There are none

3. The eccentricity is
   (A) $0$
   (B) $1$
   (C) $\sqrt{2}$
   (D) $1/\sqrt{2}$
   (E) There is none

4. The vertex is (or the vertices are) at
   (A) $(1, -3)$
   (B) $(-3, 1)$
   (C) $(1 - \sqrt{5}, -3)$ and $(1 + \sqrt{5}, -3)$
   (D) $(-3 - \sqrt{5}, 1)$ and $(-3 + \sqrt{5}, 1)$
   (E) There are none

5. The center is at
   (A) $(1, -3)$
   (B) $(-1, 3)$
   (C) $(3, -1)$
   (D) $(-3, 1)$
   (E) There is none

6. The radius is
   (A) $1$
   (B) $6/\sqrt{5}$
   (C) $\sqrt{5}$
   (D) $6$
   (E) There is none

7. The directrix (or one of the directrices) is
   (A) $y = 1 - 2/\sqrt{5}$
   (B) $x = -3 - 2/\sqrt{5}$
   (C) $y = 1 + 2/\sqrt{5}$
   (D) $x = -3 + 2/\sqrt{5}$
   (E) There is (are) none

8. The ends of the latus rectum (or one of the latera recta) are
   (A) $(1 + \sqrt{10}, -3 \pm 1/\sqrt{5})$
   (B) $(1 + \sqrt{10}, -3 \pm 1/\sqrt{5})$
   (C) $(1 - \sqrt{10}, -3 \pm 1/\sqrt{5})$
   (D) $(1 - \sqrt{10}, -3 \pm 1/\sqrt{5})$
   (E) There are none

9. The axis of parabola is
   (A) the x-axis
   (B) the y-axis
   (C) parallel to the x-axis
   (D) parallel to the y-axis
   (E) There is none

10. The parabola is concave
    (A) right
    (B) left
    (C) up
    (D) down
    (E) It is not a parabola

11. The major axis is
    (A) the x-axis
    (B) the y-axis
    (C) parallel to the x-axis
    (D) parallel to the y-axis
    (E) There is none

12. The minor axis is
    (A) the x-axis
    (B) the y-axis
    (C) parallel to the x-axis
    (D) parallel to the y-axis
    (E) There is none
13. The transverse axis is
(A) the x-axis
(B) the y-axis
(C) parallel to the x-axis
(D) parallel to the y-axis
(E) There is none

14. The conjugate axis is
(A) the x-axis
(B) the y-axis
(C) parallel to the x-axis
(D) parallel to the y-axis
(E) There is none

15. The asymptotes are
(A) \( y = x - 4 \), and \( y = -x - 2 \)
(B) \( y = x + 4 \), and \( y = -x + 2 \)
(C) \( y = x - 4 \), and \( y = -x + 2 \)
(D) \( y = x + 4 \), and \( y = -x - 2 \)
(E) There are none

II. Write the equation \( 4x^2 + 9y^2 - 56x + 54y + 241 = 0 \) in standard form and identify the conic. Then answer the questions that follow.

16. The conic is
(A) a circle
(B) a parabola
(C) an ellipse
(D) a hyperbola
(E) Degenerate

17. The focus (or one of the foci) is at
(A) \((-2, 3)\)
(B) \((-4, 3)\)
(C) \((7 + \sqrt{5}, -3)\)
(D) \((10, -3)\)
(E) There are none

18. The eccentricity is
(A) 0
(B) 1
(C) \(\sqrt{5}/3\)
(D) \(3/\sqrt{5}\)
(E) There is none

19. The vertex (or one of the vertices) is at
(A) \((-5, 3)\)
(B) \((-7, -3)\)
(C) \((7 + \sqrt{5}, -3)\)
(D) \((-7, 3)\)
(E) There is none

20. The center is at
(A) \((-3, 7)\)
(B) \((7, -3)\)
(C) \((3, -7)\)
(D) \((-7, 3)\)
(E) There is none

21. The radius is
(A) 1
(B) 2
(C) 3
(D) 6
(E) There is none

22. The directrix (or one of the directrices) is
(A) \(x = 4\frac{5}{\sqrt{5}} - 35\)
(B) \(x = \frac{4\sqrt{5} - 35}{5}\)
(C) \(x = 9\sqrt{5} + 35\)
(D) \(1\)
(E) There are none

23. The ends of the latus rectum (or one of the latera recta) are
(A) \((7 + 3, -3 \pm \frac{3}{2})\)
(B) \((-7 + \sqrt{5}, 3 \pm \frac{1}{2})\)
(C) \((7 + \sqrt{5}, -3 \pm \frac{1}{2})\)
(D) \((-7 + 3, 3 \pm \frac{1}{2})\)
(E) There are none
24. The axis of parabola is
(A) the x-axis
(B) the y-axis
(C) parallel to the x-axis
(D) parallel to the y-axis
(E) There is none

25. The parabola is concave
(A) right
(B) left
(C) up
(D) down
(E) It is not a parabola

26. The major axis is
(A) the x-axis
(B) the y-axis
(C) parallel to the x-axis
(D) parallel to the y-axis
(E) There is none

27. The minor axis is
(A) the x-axis
(B) the y-axis
(C) parallel to the x-axis
(D) parallel to the y-axis
(E) There is none

28. The transverse axis is
(A) the x-axis
(B) the y-axis
(C) parallel to the x-axis
(D) parallel to the y-axis
(E) There is none

29. The conjugate axis is
(A) the x-axis
(B) the y-axis
(C) parallel to the x-axis
(D) parallel to the y-axis
(E) There is none

30. The asymptotes are
(A) \( y = \pm \frac{5}{2}x \)
(B) \( y = \frac{2}{3}x - \frac{22}{3} \) and \( y = -\frac{3}{2}x + \frac{15}{2} \)
(C) \( y = \pm \frac{5}{2}x \)
(D) \( y = \frac{2}{3}x - \frac{22}{3} \) and \( y = -\frac{3}{2}x + \frac{15}{2} \)
(E) There are none

III. Find the equation of the parabola with vertex at \((-3, 0)\) and focus at \((5, 0)\). Put your answer in the form \(Ax^2 + Cy^2 + Dx + Ey + F = 0\) and answer questions 31 and 32 below.

31. In the equation above, the sum of the coefficients \(A + D = \)
(A) \(-10\)
(B) \(-20\)
(C) \(-32\)
(D) \(16\)
(E) \(1\)

32. In the equation above, \(C + E + F = \)
(A) \(-7\)
(B) \(-95\)
(C) \(-47\)
(D) \(64\)
(E) \(-59\)

IV. Find the equation of the ellipse centered at the origin with eccentricity \(\frac{1}{2}\) and major axis on the x-axis of length 8. Put your answer in the form \(Ax^2 + Cy^2 + Dx + Ey + F = 0\) and answer questions 33 and 34 below.
# FINAL EXAM SUMMARY

## Part 1 - The Line

<table>
<thead>
<tr>
<th>Student ID</th>
<th>I</th>
<th>II</th>
<th>III</th>
<th>IV</th>
<th>V</th>
<th>VI</th>
<th>VII</th>
<th>VIII</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>.9</td>
<td>3</td>
<td>12.5</td>
<td>12.5</td>
<td>12.5</td>
<td>0</td>
<td>0</td>
<td>49.5</td>
</tr>
<tr>
<td>2</td>
<td>12.5</td>
<td>18</td>
<td>6</td>
<td>12.5</td>
<td>12.5</td>
<td>12.5</td>
<td>9</td>
<td>11</td>
<td>94</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>15</td>
<td>6</td>
<td>12.5</td>
<td>12.5</td>
<td>12.5</td>
<td>12</td>
<td>11</td>
<td>81.5</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>9</td>
<td>3</td>
<td>12.5</td>
<td>12.5</td>
<td>0</td>
<td>12</td>
<td>9</td>
<td>46</td>
</tr>
<tr>
<td>5</td>
<td>12.5</td>
<td>15</td>
<td>6</td>
<td>12.5</td>
<td>12.5</td>
<td>12.5</td>
<td>0</td>
<td>0</td>
<td>71</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>12</td>
<td>6</td>
<td>12.5</td>
<td>12.5</td>
<td>12.5</td>
<td>9</td>
<td>0</td>
<td>64.5</td>
</tr>
<tr>
<td>7</td>
<td>12.5</td>
<td>12</td>
<td>3</td>
<td>12.5</td>
<td>12.5</td>
<td>0</td>
<td>12</td>
<td>9</td>
<td>61.5</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>12</td>
<td>0</td>
<td>12.5</td>
<td>0</td>
<td>12.5</td>
<td>0</td>
<td>0</td>
<td>37</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>9</td>
<td>3</td>
<td>12.5</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>30.5</td>
</tr>
<tr>
<td>10</td>
<td>12.5</td>
<td>12</td>
<td>6</td>
<td>12.5</td>
<td>12.5</td>
<td>12.5</td>
<td>6</td>
<td>0</td>
<td>74</td>
</tr>
<tr>
<td>11</td>
<td>12.5</td>
<td>15</td>
<td>6</td>
<td>12.5</td>
<td>12.5</td>
<td>12.5</td>
<td>6</td>
<td>0</td>
<td>77</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>9</td>
<td>6</td>
<td>12.5</td>
<td>12.5</td>
<td>12.5</td>
<td>6</td>
<td>0</td>
<td>58.5</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>12</td>
<td>3</td>
<td>12.5</td>
<td>12.5</td>
<td>12.5</td>
<td>0</td>
<td>0</td>
<td>52.5</td>
</tr>
<tr>
<td>14</td>
<td>12.5</td>
<td>9</td>
<td>0</td>
<td>12.5</td>
<td>0</td>
<td>12.5</td>
<td>9</td>
<td>0</td>
<td>55.5</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>15</td>
<td>6</td>
<td>12.5</td>
<td>12.5</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>52</td>
</tr>
<tr>
<td>16</td>
<td>12.5</td>
<td>9</td>
<td>3</td>
<td>12.5</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>43</td>
</tr>
<tr>
<td>17</td>
<td>0</td>
<td>18</td>
<td>6</td>
<td>12.5</td>
<td>12.5</td>
<td>12.5</td>
<td>0</td>
<td>0</td>
<td>61.5</td>
</tr>
<tr>
<td>18</td>
<td>0</td>
<td>12</td>
<td>6</td>
<td>12.5</td>
<td>12.5</td>
<td>12.5</td>
<td>9</td>
<td>0</td>
<td>64.5</td>
</tr>
</tbody>
</table>

## Part 2 - Conics

<table>
<thead>
<tr>
<th>Student ID</th>
<th>I</th>
<th>II</th>
<th>III</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>25</td>
<td>14</td>
<td>0</td>
<td>39</td>
</tr>
<tr>
<td>2</td>
<td>25</td>
<td>21</td>
<td>0</td>
<td>46</td>
</tr>
<tr>
<td>3</td>
<td>22</td>
<td>25</td>
<td>12</td>
<td>59</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>15</td>
<td>0</td>
<td>35</td>
</tr>
<tr>
<td>5</td>
<td>22</td>
<td>6</td>
<td>6</td>
<td>34</td>
</tr>
<tr>
<td>6</td>
<td>22</td>
<td>9</td>
<td>0</td>
<td>34</td>
</tr>
<tr>
<td>7</td>
<td>25</td>
<td>9</td>
<td>0</td>
<td>34</td>
</tr>
<tr>
<td>8</td>
<td>24</td>
<td>6</td>
<td>0</td>
<td>30</td>
</tr>
<tr>
<td>9</td>
<td>17</td>
<td>12</td>
<td>0</td>
<td>29</td>
</tr>
<tr>
<td>10</td>
<td>25</td>
<td>18</td>
<td>6</td>
<td>49</td>
</tr>
<tr>
<td>11</td>
<td>25</td>
<td>11</td>
<td>6</td>
<td>42</td>
</tr>
<tr>
<td>12</td>
<td>24</td>
<td>20</td>
<td>0</td>
<td>44</td>
</tr>
<tr>
<td>13</td>
<td>25</td>
<td>18</td>
<td>0</td>
<td>43</td>
</tr>
<tr>
<td>14</td>
<td>25</td>
<td>15</td>
<td>6</td>
<td>46</td>
</tr>
<tr>
<td>15</td>
<td>25</td>
<td>16</td>
<td>0</td>
<td>41</td>
</tr>
<tr>
<td>16</td>
<td>22</td>
<td>10</td>
<td>0</td>
<td>32</td>
</tr>
<tr>
<td>17</td>
<td>22</td>
<td>9</td>
<td>12</td>
<td>43</td>
</tr>
<tr>
<td>18</td>
<td>25</td>
<td>14</td>
<td>0</td>
<td>39</td>
</tr>
</tbody>
</table>

## Possible Score

|   | 12.5 | 18 | 6 | 12.5 | 12.5 | 12.5 | 15 | 11 | 100 | 25 | 25 | 12 | 62 | 162 |

## Class Average

|   | 5   | 12 | 4.3 | 12.5 | 9   | 10 | 6  | 1  | 59.8 | 22 | 13 | 3  | 38 | 98  |

## Related Sections of Text

|   | 2.1  | 2.2  | 3.1  | 3.1  | 3.2  | 3.2  | 4.6 | 4.6 | 4.2 | 4.2 | 4.6 | 4.6 | 4.6 | 4.6 |

## Exam Total

|   | 88.5 | 140 | 140.5 | 81  | 95.5 | 95.5 | 67  | 67  | 123 | 119 | 102.5 | 98  | 84  |

## Possible Score Total

|   | 100 | 25 | 25 | 12 | 62 | 162 |
### Evaluation Sheet

**COMPUTERIZED ANALYTIC GEOMETRY**

Student: _______

I. Do you feel you have learned more using the computerized approach than the normal classroom approach?

<table>
<thead>
<tr>
<th>More</th>
<th>Less</th>
<th>No difference</th>
<th>Comment</th>
</tr>
</thead>
</table>

II. Do you enjoy the overall computerized approach?

<table>
<thead>
<tr>
<th>Yes</th>
<th>No</th>
<th>Indifferent</th>
<th>Comment</th>
</tr>
</thead>
</table>

III. Did you spend more time than normal on this course during the computerized portion?

<table>
<thead>
<tr>
<th>More</th>
<th>Less</th>
<th>No difference</th>
<th>Comment</th>
</tr>
</thead>
</table>

IV. Assuming you learned something, determine the percentage each of the following activities contributed to your learning.

<table>
<thead>
<tr>
<th>Activity</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classroom lecture</td>
<td>A. Quizzes</td>
</tr>
<tr>
<td>Reading textbook</td>
<td>5. CAI</td>
</tr>
<tr>
<td>Programs</td>
<td>Sum 100%</td>
</tr>
</tbody>
</table>

V. Was the content of the course too difficult?

<table>
<thead>
<tr>
<th>Yes</th>
<th>No</th>
<th>If at a slower pace</th>
<th>Comment</th>
</tr>
</thead>
</table>

VI. Were the required programs too difficult?

<table>
<thead>
<tr>
<th>Yes</th>
<th>No</th>
<th>If I knew more programming</th>
<th>Comment</th>
</tr>
</thead>
</table>

VII. Was the CAI useful?

<table>
<thead>
<tr>
<th>Yes</th>
<th>No</th>
<th>Didn't try any</th>
<th>Comment</th>
</tr>
</thead>
</table>

VIII. Were the quizzes too difficult?

<table>
<thead>
<tr>
<th>Yes</th>
<th>No</th>
<th>If at a slower pace</th>
<th>Comment</th>
</tr>
</thead>
</table>

IX. Would you like any of your future math courses taught this way?

<table>
<thead>
<tr>
<th>Yes</th>
<th>No</th>
<th>Indifferent</th>
<th>Comment</th>
</tr>
</thead>
</table>

X. If you could redesign the computerized approach, how would you do it?

[Blank space for comments]

XI. Was the final exam representative of what you learned?

<table>
<thead>
<tr>
<th>Yes</th>
<th>Somewhat</th>
<th>No</th>
</tr>
</thead>
</table>

XII. Overall comments.

[Blank space for comments]
### EVALUATION SUMMARY (Part 1)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>More</td>
<td>Less</td>
<td>No</td>
<td>Dif</td>
<td>Yes</td>
</tr>
<tr>
<td>1</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>35</td>
<td>40</td>
</tr>
<tr>
<td>2</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>3</td>
<td>90</td>
</tr>
<tr>
<td>3</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>16</td>
<td>25</td>
</tr>
<tr>
<td>4</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>75</td>
<td>15</td>
</tr>
<tr>
<td>5</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>50</td>
<td>35</td>
</tr>
<tr>
<td>6</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>7</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>35</td>
<td>35</td>
</tr>
<tr>
<td>8</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>40</td>
<td>30</td>
</tr>
<tr>
<td>9</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>10</td>
<td>55</td>
</tr>
<tr>
<td>10</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>30</td>
<td>60</td>
</tr>
<tr>
<td>11</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>35</td>
<td>45</td>
</tr>
<tr>
<td>12</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>45</td>
<td>45</td>
</tr>
<tr>
<td>13</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>14</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>15</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>16</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>17</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>20</td>
<td>50</td>
</tr>
<tr>
<td>Total</td>
<td>1</td>
<td>8</td>
<td>8</td>
<td>9</td>
<td>3</td>
</tr>
</tbody>
</table>

* a averages instead of total
EVALUATION SUMMARY (Part 2)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Yes</td>
<td>No</td>
<td>No, if</td>
<td>Yes</td>
<td>No</td>
<td>Did'nt try</td>
</tr>
<tr>
<td>1</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>12</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>13</td>
<td>4</td>
</tr>
</tbody>
</table>

<sup>a</sup> Attitude index is the sum of the responses to questions 1, 2, and 9 where +1 is a positive response, i.e., learned more, enjoyed course, like future courses taught this way; 0 is a neutral response; and -1 is a negative response; ex: Attitude index for Student 15 = (-1) + (+1) + (0) = 0
F.3 Student Comments

Subject Matter

"The content at the end was quite difficult and class lecture did not help much." [2]¹.

"Analytic geometry concepts were not, however, covered too quickly." [3]

"[Content] not really too difficult as far as math goes - computer made it fairly difficult." [7]

"There were a few sections that were fairly difficult." [13]

"For an introduction into analytic geometry - good." [13]

"Maybe it, [the content] was, [too difficult] and the speed didn't help any." [15]

"Some parts of the text were hard to understand when not covered in class." [16]

Course Length and Difficulty

"The course was so fast, I found myself constantly trying to keep up." [1]

"The course should be run over at least a period of a semester." [1]

"I think a four week course is much too short. Slower pace is needed." [1]

"It was too time consuming and impossible to stay up with assignments." [2]

"Moreover, the great amounts of work required discouraged the non-programmers from learning anything about programming." [3]

"Since the whole program was run so quickly, I feel I've learned less." [5]

"One had to do all or nothing and I didn't have time to do all." [5]

¹Number in brackets is student identification number.
"Between the programming and general studying it was difficult to learn anything except the general facts." [6]

"The course, I believe, was basically a good one. I do feel that making it a full eight week course would improve it immensely. It would make it so the student would only have to write 1 program a night. Only half as much reading each night would be required. At this slower pace the student would be able to completely take in all material. This would also allow an additional 10-15 minutes a day in class for programming instructions on the program for that day." [7]

"It was enjoyable working on the computer, but the pace was too fast to enjoy the program." [9]

"I think the course should be at least 6 and possible 8 weeks. So that the teacher could get into the subjects more deeply and completely. Thus Mr. Klitz barely got through the material once, if that, and had no time to explain it well." [9]

"I think they [future computer math courses] would have to be taught at a slower pace." [9]

"I do not like the approach because of the time involved - I am rather busy and cannot find an hour for programming each day." [12]

"The pace was too fast." [12]

"I think that I would have enjoyed the approach had it not been so hurried. In math, I have still not reached the point of proficiency where I can sit down and work great volumes of problems in time for the next day's class." [12]

"I think the most important change that could be made would be if the program were lengthened to perhaps eight weeks, cutting the workload in half. With this addition, I think that the course would be very valuable." [12]

"I believe that at a slower pace, there would have been more understanding of the very difficult sections of the text such as cosines, circle, ellipse, etc." [14]

"Too fast on covering material. If this course had been given over the whole year, I think much more would have been learned and more people would have tried the programming, CAI's, and spent more time on homework." [15]
"I would take more time in teaching the course." [16]

"I should not try to teach high schoolers at twice the rate suggested for the smartest college student." [17]

Individuality

"The ability for individuals to work independently throughout the program was one of its biggest assets. It was also possible to pick out one area (quizzes, programming or CAI) and place the emphasis on the one of these areas the student preferred." [6]

"I like the independence of the approach, but the computer helped little." [11]

"The independence [was] too great." [12]

"The course is plagued with one great problem - the ease of doing nothing. I am perhaps in a minority in taking advantage of this, but I was rather negligent the entire time." [12]

Integration

"Classroom work was not connected well enough with computer part of course." [3]

"Make it more geometry and less computer. The hardest sections were never discussed well, so if that night you still didn't understand it after reading it, you flunked the quiz." [3]

"I think that few students got much more out of the course than just an introduction to analytic geometry. More time is needed to explore the development of the geometry. For example, a person who is given time to look into the subject and derive his own formulas will be given a greater understanding of the subject, and the formulas and the whole system will mean something to him. Thus he will remember them, and have founded a solid basis for further work in analytic geometry." [5]

"In conclusion I would like to say that when the student has a full comprehension of the classroom part of the course, the computer part would actually be easy, and maybe even enjoyable. Therefore I suggest that the computer part be secondary to the classroom part." [5]

"The program should be more closely related to problems in the book." [6]
"The program was different and a challenge, but I don't really think it succeeded in teaching a great deal of analytic geometry. It was a good way to teach people to operate the computer. The large amount of time required in programming and card writing makes it the major part of the course - the text becomes a reference book to help figure out procedures for computing answers." [6]

"Less time is spent on learning the math. A lot of time is spent in programming, running, etc." [7]

"I really feel that I can learn more math if I spend 100% of work on mathematics." [7]

"I don't know anything about computers so I felt I was at a loss in the course. I don't enjoy working with the computer. I would make sure everyone knew much more about the computer. Only some people knew enough to make it worthwhile in this class." [8]

"The computer part of the program came to be a nuisance and what I learned was only from class and myself studying at home." [11]

"Classroom time should (have) been used going over programs." [11]

"Classes shouldn't be mandatory except for quizzes. Either make computers more important or not at all. At present little is learned with computers." [11]

"Perhaps another reason for my laziness is that the computer is not a novelty. I am not terribly anxious to use is simply because it exists, and the programs you asked for were rather dull for me (I haven't looked at the later ones, perhaps they were better)." [12]

"The computer did not play a very important part in my learning the course. But it would be good for extra-activity." [13]

"The inclass course is OK." [13]

"I feel I learned about the same amount of knowledge, in either [computer or normal] teaching manner. The interaction between computer and student was a stimulating experience." [14]

"I thought that this course was definitely successful in the area of learning analytic geometry but unsuccessful in the computer phase of the program." [14]
"It's a new fun way of learning." [15]

"Analytic geometry was interesting but the computer was trouble." [16]

"A good idea - too many problems when put into practice."
[17]

"The class using the computer/[should be] very small in numbers (6 to 8)." [17]

Computer Availability

"To get on the computer was not difficult because few people were doing the programs. If everyone, however, in the class did programs, I am sure it would be difficult to run a program." [2]

"However, I can envision a major source of trouble if everyone was trying to run programs during just 1 or 2 periods a day. It was hard enough to get on when just 4 or 5 were running and correcting programs." [3]

"The computer was hardly ever open, so it was difficult to get on it for the time you needed it. Somehow get more time for everyone to use the computer." [4]

"Of course a person should be required to do it [computer part of course], but crowds at the computer prevent this from being feasible." [5]

"Another major problem could have been computer usage. A great amount of time is needed by each student on the computer. A CAI might take 10 minutes, a program 5 or so minutes for each try. If a large number of students had continued working on the program it would have been impossible to get on the computer for more than 5 or 10 minutes a day." [6]

"And evenly distribute computer time. This would give everybody an equal chance. Perhaps 10 minutes a day per student could be allotted." [7]

"One computer was not enough for this course. A small clique thought the computer was theirs and theirs alone. To run a CAI you had to beat those guys off or stay after school. I didn't want to waste my time with those people and didn't particularly want to say after school. So I did little in the way of CAI's." [8]
"Also computer time per person should be limited to say 10-20 minutes a day so that certain people could not take over the computer and shut out the people that need help." [9]

"Didn't use the computer because about 4 or 5 students monopolized it." [10]

"As far as computers are concerned a sign-up sheet should be used. Students sign up for allotted amounts of time. This prevents a small group of students from dominating the computer. [13]

"I enjoyed the course in that I was interested in analytic geometry. I would've enjoyed it more if I had been able to work on the computer - Some people seemed to have a monopoly on the computer, this made it harder to get in and run CAI's." [16]

Operational Procedures

"In instances when a person runs a program again after making corrections, the computer should not ask for the student number again; when a different student runs the program, he has to feed the check program in again because his line numbers do not necessarily correspond to the previous student's anyway." [2]

"Furthermore, no one bothered to take the trouble to sign in when running programs. It was a pain, especially if you were in the computer room most periods." [3]

"Furthermore, in the future, I would hope that the student would not have to be graded on the computer section of the course." [5]

"A final problem I observed was that of "fixing" programs. By typing in "150" and "b85 PRINT" it was possible to have an unlimited number of trys. It was also possible to fix CAI tries by stopping anywhere before the end of the program. Also, it was possible (for one student in particular) to buy or borrow other peoples programs to glorify his record." [6]

Quizzes

"The quizzes were excellent means of checking one's knowledge and fit into the classroom part without taking up too much time." [3]

"Less time [should be] spent on quizzes and more spent on in-class lecture and help." [4]
"Quizzes also good." [6]

"And I think that they [the quizzes] were not helpful and too often. I think that maybe one every other day would be enough." [9]

"Some [quizzes] were [difficult]." [13]

"Quizzes helped a lot. Shouldn't count too heavily on grade though." [13]

"Possibly too much covered on each quiz." [15]

"At the end they got harder." [16]

"Quizzes should be more general than ridiculously picune." [17]

Programming

"I might have learned much more had I been a good programmer." [1]

"It would be good if each student was carefully taught to program." [1]

"One has to know well the material before writing a program." [2]

"A great deal of time goes into marking cards and running programs." [2]

"Each program in itself is not too difficult, but it is impossible to keep up with assignments." [2]

"More time should be allowed for the programs and no limit should be set on the number of tries." [2]

"In order to do the reading, write the programs and put on cards for any common night could take 2 1/2 hrs." [2]

"All check programs should be entirely checked, so they do not conflict with instructions for the program and so that all answers are correct." [2]

"Also, coefficients of equations should be the same as those in the corresponding part of the text book." [2]

"A subroutine or function to find the distance between two points would be helpful and save time." [2]
"Quite a few of the programs involved material not covered in class." [3]

"It was too time consuming to write all programs → 2 1/2 hours a night doing programs, 1 1/2 hours (2 periods) running programs, CAI's per day." [3]

"Programming much too fast." [3]

"Should have been just 1 per night, especially towards end of course." [3]

"First be sure all students can program to necessary standards for course." [3]

"A constant source of trouble were mistakes in the check programs and CAI's. I found about half a dozen check programs in error (11, 12, 19, 21,...)." [3]

"Spent lots of time debugging check programs." [3]

"Two programs are too much work to be done in one night. It would be [better] at a slower pace with, say, 2 programs due every three days." [5]

"There were just too many [programs] - a few were too difficult, but not many." [6]

"1 program per night (or maybe 3 over a 2 day period)." [6]

"There were several bugs in the checking programs and CAI, but these can only be worked out with time, I guess." [6]

"The programs sometimes got a little rough, but I enjoyed it." [7]

"I believe not enough time was spent going over programs." [7]

"I found that I spent too much time trying to write programs. The writing of programs took much more time than regular homework." [9]

"Also many of the checking programs had errors in them that would throw me off. Perhaps if there were fewer programs to write one could concentrate better on the geometry." [9]

"There were too many programs. They tended to take all of my time and I did not study much. Perhaps if there were one program a night or two every three days you would get their benefits, but not take all of your time." [9]
"Trying to write 2 programs a night would take hours - therefore I generally concentrated on quizzes and stopped writing programs." [11]

"The only major changes that I would make is shorter and more rigid assignments in programming - If I had to use a computerized approach." [12]

"The point remains that I was at the beginning of the course a reasonably proficient (relatively) programmer and a reasonably good math student. I found it difficult to motivate myself sufficiently to do the work required to do a good job on the course. This is certainly in part due to my own inadequacies, but part, also, because of the program." [12]

"The programs did not help me learn anything about analytic geometry. There was too much homework as far as writing programs. Most of them were very long and involved - much chance for mistakes." [13]

"Programs challenging, but not help much - too much work." [13]

"I believe if a more thorough programming course was taught, many of use could have done probably 75% of the required programs instead of 0% or almost none at all. Some programming was very difficult however." [14]

"One correction I believe would be to lessen the homework problems to give the student more time to spend on them." [14]

"Slow down, go into more detail on programming, with less homework and more time to have to do programs that could help. The programs could be useful, but as I hadn't done much before, they were very difficult, and took much more time." [15]

"If I had known how to program I think I would've gotten some of them." [16]

"Make sure that everyone could work with the computer before starting." [16]

"Writing a program is a challenge and is a stimulus for better, broader programs on that subject. Having to rattle off two programs a night makes programs very specific (therefore relatively useless), leaves no time to improve on programs and gives programming an air of production line anonymity." [17]
Terminal Presented Instruction

"[Useful] but didn't learn much from them." [3]

"Many CAI's had errors too." [3]

"On CAI's a constant source of bother was the fact that words were not printed out in full, making one spend half his time deciphering the question." [3]

"The CAI's should be revised to more test the students understanding of geometry, not just skim the surface." [5]

"They didn't really teach anything, but it was good practice doing problems." [6]

"The CAI's should be used just to give problems, not explain how to do it (unless the answer given is wrong). A student can read the same thing in the book as on the CAI's. [6]

"[The CAI's were useful] but not a great help. They should have been set up so that there was more computer response. Like if someone missed a problem, the computer would tell him how to do it and give him another problem, and if he still missed it to refer the student to a section in the book." [9]

"The CAI's were good but there were too many problems there. If a student did not understand a part of it, he is not going to be able to learn by trying to work problems. [9]

"I noted of CAI's, there should be two sets of programs. One a student goes to for problems, like the CAI's, and another which explains things to the student." [9]

"I only tried a couple but they were helpful." [11]

"The computerized program is in good shape, especially with the use of CAI's." [14]

"They [CAI] can help you with basic problems." [15]

Final Exam

"The final exam just on conic sections and the straight line. We spend a lot of time on other areas, too, for example polar coordinates." [1]

"The course was fun, the instruction good, the quizzes also good, but the final test was not. It didn't even discuss polar coordinates, any set theory, or the 3-D coordinate system. 1/2 the test was basically 2 problems involved with conic sections." [6]

"The final exam concentrated almost entirely on conics. Specific knowledge was needed. This was only one chapter out of six, there should have been more from the others." [8]

"[The final exam had] too great a stress on the conic sections for the time we spent on them." [9]

"Final exam was mainly based on conic sections." [10]

"We learned, supposedly, much more than what was basically on the final exam." [14]

"Most of the final exam covered conics; not much on three dimensional coordinate system, polar coordinates, etc." [16]
Computerized Analytic Geometry

I. Do you feel you learned more using the computerized approach than the normal classroom approach?  
More □ Less □ No difference □ Comment □

II. Did you enjoy the overall computerized approach?  
Yes □ No □ Indifferent □ Comment □

III. Did you spend more time than normal on this course during the computerized portion?  
More □ Less □ No difference □ Comment □

IV. Assuming you learned something, determine the percentage each of the following activities contributed to your learning.  
1. Classroom lecture 20% 4. Quizzes 10%  
2. Reading textbook 20% 5. CAI 10%  
3. Programs 30% Sum 100%  

V. Was the content of the course too difficult?  
Yes □ No □, if at a slower pace □ Comment □

VI. Were the required programs too difficult?  
Yes □ No □, if I knew more programming □ Comment □

VII. Was the CAI useful?  
Yes □ No □ didn't try any □ Comment □

VIII. Were the quizzes too difficult?  
Yes □ No □, if at a slower pace □

IX. Would you like any of your future math courses taught this way?  
Yes □ No □ Indifferent □ Comments □

X. If you could redesign the computerized approach, how would you do it?  
I think the basic design was good.  

XI. Was the final exam representative of what you learned?  
Yes □ Somewhat □ No □

XII. Overall comments.  
My overall comment would be satisfactory with the approach. I think it would have been more enjoyable if we had just done a week of lectures, a day or two, and that the course not been progressed in such a way and that you, Professor, put a few specimen problems, did not necessarily for the computerized approach.
Quiz Instructions

The class will be given a 7-item multiple-choice quiz each day except for the first and last day. This results in 18 quizzes overall. Very little calculation is necessary to correctly answer the quiz questions if the student comprehends the subject matter. However, if the student doesn't know the material very well, he may try a long, tedious, brute-force approach which could consume a large portion of the quiz-taking time (10 minutes). It is therefore recommended that no student spend more than a minute on each question and use the remaining time for checking.

Each student will receive a deck of eighteen (18) cards with the statement number, "DATA", his student number, quiz number, and appropriately spaced commas all prepunched. The student therefore only has to mark his answers on the proper card in the following format:

- answer to question 1 - card column 14
- answer to question 2 - card column 16
- answer to question 3 - card column 18
- answer to question 4 - card column 20
- answer to question 5 - card column 22
- answer to question 6 - card column 24
- answer to question 7 - card column 26
- minutes spent on homework - card columns 28-30

The quiz number will be given on top of each quiz sheet. If a student forgets his quiz cards, spares will be available; however, all prepunched columns will now have to be pencil marked.

At the end of the quiz, Mr. Klitz will collect the cards, check for improper quiz number, and review the quiz. After class the cards will be taken to the computer room where they will be input to the quiz grading program. This program will print a report for Mr. Klitz containing each student's quiz scores, program record, and CAT record. It will also store the quiz scores which will then be available when the students interact with their homework programs.

Since the results of the quiz will not affect any students course grade whatsoever (the results will be used only for recommending CAI sessions and experimental statistics) it is suggested that if a student doesn't know an answer, he should fill in the "0" position. It is possible that the correct answer is not one of the choices. It is important to remember, however, that none of the quiz columns be left blank. With regards to the "minutes spent on homework", include time spent on CAI and program interaction but please do not include time spent on snack breaks, telephone, etc.
G.2 Quiz deck setup

QUIZ DECK SETUP

RUN Card

DATA 99, ...

STUDENT QUIZ DECK

DATA 9099

CORRECT QUIZ NUMBERS CAN BE CHECKED BY HOLDING CARDS UP TO LIGHT CARD 11.

DATA 9001

DATA 9000

CORRECT QUIZ ANSWERS

STUDENT QUIZ DECK
Homework instructions 281

Computerized Analytic Geometry

Programming Instructions

Each student's homework for the course will consist of two programming problems each night. This will result in 36 programs overall. These programs are to be written in BASIC and usually should not require more than 20 BASIC statements. The problem definition sheets give statement number limits for each program. These limits protect the data generation and result checking portions of the overall program resident in the computer and also allows the student to use previous programs as subroutines. Because of the data generation and result checking portions of the program, the student need not be concerned with any input/output however, he must be very careful in maintaining the required variable names.

It is unlikely that a student will get a program correct the first time and therefore should continue debugging the program until he gets correct answers, of which he will be notified by the computer. However, there may be a tendency to get behind if too much time is spent on one program; therefore the checking program will only allow a student 7 tries. It is recommended that if a student gets behind, he only do one new program each day until he can catch up. Those students who are having no difficulty with the programming are encouraged to work as far ahead as they can go. (assuming they take the recommended CAI sessions while they are working ahead.)

When a student has written a program and coded it on mark-sense cards, he will take it to the computer room, sign in on a "sign-in sheet", and ask the operator to load "program ___ check tape." The operator will then load the student's program deck and, assuming no compilation errors, the student will sit down at the teletype to answer the question "STUDENT NO.?" If the number typed in is a valid student number, the student's program will execute several sets of data supplied by the checking program. The results will be checked and, if correct, "PROGRAM CORRECT" will be printed. If the answers are incorrect, the program will print out the data used and the correct answer to help in the debugging process. Even though no output is necessary, it may be judicious to print out a few intermediate results of each program.

After the program has been checked, the student should ask the operator to load the "STUDENT EXECUTIVE TAPE". After this program is loaded, the student will be asked the following questions:

1. "WOULD YOU LIKE YOUR QUIZ SCORE?" (type 1 for yes)
2. If yes, "FOR QUIZ NO.?"
3. The program will respond with "SCORE IS ____"
4. "WOULD YOU LIKE YOUR RECORD TO DATE?" (type 1 for yes and 0 for no)
5. The program will respond with all the students quiz scores, program tries, and CAI results to date.
A set of ten (10) computer-assisted-instruction (CAI) programs will be available to the students. These programs are designed to furnish a quick review for the student who isn't progressing very satisfactorily, or a brief introduction to a new area that hasn't yet been covered during the classroom lecture for those students who are working ahead. It is not anticipated that every student will take every CAI session. When a student interacts with the "Student Executive Program" after his homework program has been checked, he will be told whether to take a CAI session or not. This decision will be based on his current quiz score, his CAI record, and his programming record.

The CAI programs consist of a series of questions that a student is expected to answer. If the student answers incorrectly, he is given the correct answer with a short explanation and is then asked the same question again but with different data. Occasionally the program will recycle two or three questions when an incorrect answer is given. When numeric questions are not required, multiple-choice or true-false questions will be asked. The convention of "1" for true and "0" for false will be used throughout. Please remember to hit the return key after answering a question.

Each program will have at most ten (10) questions and each session should last no longer than 5 minutes. Upon completion, the total number of incorrect answers will be printed out. A student may retake the same CAI program up to a limit of three times.

To interact with a CAI program the student asks the operator in the computer room to load CAI tape number ___ after signing in on the "sign-in" sheet. The student then sits down at the teletype and answers the questions asked of him. To eliminate an infinite number of questions; a student will automatically be moved to the next question if he misses the same question three (3) times.
There will be four separate activities of this computerized course as follows:

I. DAILY PRINTOUT OF STUDENT DATA.

The first activity each morning will be the running of the DUMP routine. This routine will punch out all the student record data for possible future recovery in case of a system breakdown. Since the last statement of the program is a PLIST, the paper tape punch must be turned on.

II. OPERATING THE QUIZ SCORING PROGRAM.

This is a two-tape program. When Mr. Klitz brings the quiz cards to the computer room, Tape A should be loaded before the quiz cards are loaded into the card reader. Tape A will have a CARD statement as its last statement and the quiz deck will have a RUN card as its last statement. After the quiz results are printed, the operator should load Tape B which will automatically print out the remainder of the report. The cards should not be reloaded during this phase.

III. OPERATING THE PROGRAM CHECKOUT PROGRAMS.

There are 38 programs to be written by each student. These programs will be checked by a series of data generation and result checking program. These checking programs will be stored on 38 separate paper tapes. When a student wants a program checked he will ask the operator to load the check tape for that program. After the check tape is loaded, the operator will load the student's program (written on mark sense cards) through the card reader. If the student's program compiles correctly, the student will then interact with the check program at the teletype. Please remind the students to sign in and out on the "sign in" sheet. After the student has had his program checked, he may wish to look at his record to date. This will require the loading of the STUDENT RECORD program which also requires some student interaction at the teletype.

IV. OPERATING THE CAI PROGRAMS.

There are 10 CAI programs which will require extensive student interaction. When a student wishes to interact with a CAI program, he will ask the operator to load CAI tape no. The student will then sit at the teletype to interact with the program. Please remind the students to sign in and out on the "sign in" sheet.
<table>
<thead>
<tr>
<th>STUDENT</th>
<th>PROGRAM NO</th>
<th>CAL. NO.</th>
<th>TIME IN</th>
<th>TIME OUT</th>
<th>COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>