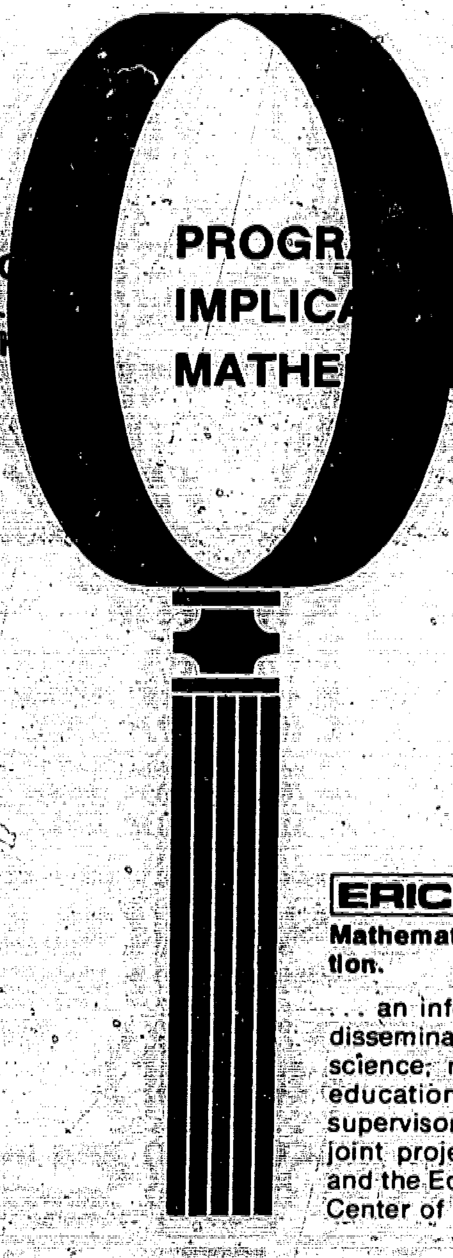
DOCUMENT RESUME

ED 200 436 SE 034 580

AUTHOR	Spikell, Mark A., Ed.
TITLE	Programmable Calculators: Implications for the Mathematics Curriculum.
INSTITUTION	ERIC Clearinghouse for Science, Mathematics, and Environmental Education, Columbus, Ohio.
SPONS AGENCY	National Inst. of Education (ED), Washington, D.C.
PUB DATE	Dec 80
CONTRACT	400-78-0004
NOTE	124p.
AVAILABLE FROM	Information Reference Center (ERIC/IRC), The Ohio State Univ., 1200 Chambers Rd., 3rd Floor, Columbus, OH 43212 ($3.00).
EDRS PRICE	MF01/PC05 Plus Postage.
DESCRIPTORS	*Calculators; Discovery Learning; Educational Technology; Instructional Materials; Lesson Plans; *Mathematical Enrichment; Mathematics Education; *Mathematics Instruction; *Problem Solving; *Programing; Secondary Education; *Secondary School Mathematics
IDENTIFIERS	*Programable Calculators

ABSTRACT

This document is a collection of reports presented at a programable calculator symposium held in Seattle, Washington, in April, 1980, as part of the annual meeting of the National Council of Teachers of Mathematics (NCTM). The session was designed to review whether the programable calculator has a place in the school mathematics program, in light of the current availability of the microcomputer. The presentations at the symposium supported the view that such calculators do have a role to play in the curriculum, and the collected papers of the contributors provide ample evidence of the many ways programable calculators can be used. In addition to the presented papers, two other contributions solicited by the editor to enhance the usefulness of this work to educators are included. (MP)



PROGR ABLE CALCULATORS IMPLICA NS FOR THE MATHE CS CURRICULUM

Mark A. Spikell
Editor

ERIC® Clearinghouse for Science, Mathematics, and Environmental Education.

... an information center to organize and disseminate information and materials on science, mathematics, and environmental education to teachers, administrators, supervisors, researchers, and the public. A joint project of The Ohio State University and the Educational Resources Information Center of NIE.

PROGRAMMABLE CALCULATORS

IMPLICATIONS FOR THE MATHEMATICS CURRICULUM

Mark A. Spikell
Editor

ERIC Clearinghouse for Science, Mathematics
and Environmental Education
The Ohio State University
1200 Chambers Road, Room 310
Columbus, Ohio 43212

December, 1980

TABLE OF CONTENTS

This publication was prepared with funding from the National Institute of Education, U.S. Department of Education under contract no. 400-78-0004. The opinions expressed in this report do not necessarily reflect the positions or policies of NIE or U.S. Department of Education.

PREFACE

The latter part of the decade of the '70s was characterized by some important technological and business developments with potential for significant impact on the school mathematics curriculum at all levels. On the technological front, advanced calculating devices, in the form of hand-held programmable calculators, were developed and manufactured with capabilities that blurred the distinction between what is a calculator and what is a computer. As a result, calculations and problem-solving tasks previously relegated to computers, because they were too time-consuming or complex to perform, became possible for students of all ability levels to handle. On the business front, the cost of these programmable calculators decreased rapidly as production increased. By the end of the decade one could purchase a fully programmable hand-held calculator for as little as $40. Because of the low cost, portability, and increasing availability of these machines, some educators began to consider the possible impact that programmable calculators might have on the teaching of mathematics. Such considerations served as the impetus for the preparation of this monograph.

In the winter of 1978, a review of available literature on the programmable calculator revealed relatively few contributions discussing the role of the programmable calculator in school mathematics. In the spring of 1979, proposals were being solicited for sessions by the program committee for the April 1980 annual meeting of the National Council of Teachers of Mathematics to be held in Seattle, Washington. A proposal was presented pointing out the dearth of information available on the role of programmable calculators in school mathematics, suggesting the potential impact these machines might have on the curriculum, and requesting that a special session be held at the annual meeting where interested persons could share ideas. The program committee, chaired by Richard Lodholz of the Parkway School District in Chesterfield, Missouri, accepted the suggestion and committee member James M. Rubillo of Bucks County Community College in Newtown, Pennsylvania invited this writer to organize and moderate such a session.

The session had several unique features and the National Council of Teachers of Mathematics should be applauded for its willingness to offer an experimental session at an annual meeting. The session, entitled "The Programmable Calculator--A Tool for the 1980s," was specifically designed as a symposium for persons interested in sharing ideas on the role of programmable hand-held calculators in school mathematics. The format of the session provided for opening remarks by the moderator and eleven five-minute talks by presenters, with at least one-half hour devoted to questions, answers, and discussion from the audience.

While some possible contributors were contacted by the moderator, it is of interest to note that the description of the session in the program booklet included an invitation to persons wishing to contribute to send copies of proposed talks to the moderator. Actual contributors were then selected by a review process and several of those ultimately chosen were persons who responded to the program booklet invitation.

When contributors were notified that their proposed talks had been selected for presentation, they were advised that the results of the symposium might be published as a collection of papers. Hence, a condition of participation was the individual's willingness to submit a formal paper so that the ideas shared at the symposium might be more widely disseminated at some future date. After the symposium, a proposal with first drafts of papers was sent to ERIC/SMEAC at The Ohio State University in order that the collection might be considered for publication. Following a review process, a favorable decision was made in the fall of 1980 to publish this monograph.

In addition to the papers presented at the symposium, this monograph includes two other contributions solicited by the editor to enhance the usefulness of this work to educators. Professor Gerald R. Rising, a well-known and respected mathematics educator at the State University of New York at Buffalo, who has done extensive work with programmable calculators, was invited to prepare an introductory article for the monograph giving a setting for the potential curriculum applications of the programmable calculator. Also, Ms. Jill Coup, a librarian at George Mason University, was invited to help the editor prepare a classified and partially annotated bibliography of references on programmable calculators. Symposium participants are grateful to Professor Rising and Ms. Coup for accepting the invitations and providing their respective fine contributions.

In the preparation of any manuscript for publication there are always many persons whose assistance is invaluable. Clearly, this monograph would not have been possible without the cooperation and fine work of the symposium contributors and the support of others already mentioned. But there remain three persons who deserve special mention. To Professor Marilyn N. Suydam, Associate Director for Mathematics Education at the ERIC Clearinghouse for Science, Mathematics and Environmental Education, and one of our profession's true leaders, many thanks for her valuable support and assistance in the preparation of this monograph. To my good friend and frequent collaborator, Professor Stephen L. Snover of the University of Hartford, many thanks for his help in all phases of this project. Every editor needs a colleague like Steve to call upon for those unexpected and last-minute crises. And, finally, my thanks, appreciation, and love to my wife Laurie whose understanding knows no limit. At times when she deserved more of my attention, she willingly sacrificed so that I might be involved in this project.

Dr. Mark A. Spikell
Department of Education
George Mason University
Fairfax, VA 22030

FOREWORD

In opening remarks at the programmable calculator symposium in Seattle, I chose to set the stage for the contributors by providing my answer to the question, "In light of the microcomputer, does the programmable calculator have a place in the school mathematics program?" The question needed addressing because by 1980 some educators were suggesting that programmable calculators would have little or no impact on the curriculum. These educators seemed to believe that the dramatic technological and business developments in the manufacture of microcomputers--increasing machine capabilities and rapidly decreasing costs--would enable schools to have these devices in large numbers. Consequently, there would be little need for or advantage in having programmable calculators available for students to use as tools in the study of mathematics. The thinking seemed to be, why have the limited problem-solving capability of programmable calculators when the virtually unlimited potential of the microcomputer was available and at reasonable costs.

Few would dispute the potential impact that microcomputers may have on the school curriculum. But any suggestion that programmable calculators have little to contribute is most short-sighted. Even as microcomputers proliferate and become more widely available in schools, programmable calculators have a great deal to offer. Instructionally, these calculators are effective devices for doing much of what can be done by microcomputers. Consider the following list of instructional applications for which microcomputers might be used. While the list is not exhaustive, it does cover some of the more obvious applications.

1. to provide drill and practice experiences
2. to serve as an independent study tool
3. to perform problem-solving tasks
4. to develop programming skills
5. to permit simulations
6. to conduct testing
7. to perform computer-managed instructional activities
8. to do data analysis
9. to function as an information retrieval device
10. to provide word processing capabilities
11. to offer computer literacy information
12. to explore gaming experiences

Many of the twelve instructional applications cited are not restricted to microcomputers. Programmable calculators can readily be used for at least eight of the twelve. They are:

1. to provide drill and practice experiences
2. to serve as an independent study tool
3. to perform problem-solving tasks
4. to develop programming skills
5. to permit simulations

6. to do data analysis
7. to provide computer literacy information
8. to explore gaming situations

What's more, programmable calculators offer some advantages over microcomputers. The calculators are so inexpensive that virtually every student could be supplied with one. A classroom set of programmables--enough, say, for each student in a class of 30-- costs no more than perhaps two or three microcomputers. Also, the programmable calculator has the superb feature of portability. As a hand-held device it can be easily carried wherever the user needs or wants to use it. Finally, since the calculators are (usually) battery-operated, they can be conveniently used even in places where access to electrical outlets is not available.

In summary, the view I shared in opening the symposium was that programmable calculators do have a role to play in the school mathematics curriculum. The presentations at the symposium supported this view and the collected papers of the contributors provide ample evidence of the many ways programmable calculators can be used in the study and teaching of mathematics.

In concluding this foreword, I provide for interested readers a brief summary of the central ideas shared by each author in the articles presented in this monograph.

Rising in his lead article addresses the question of whether or not the programmable calculator will be of only passing interest in the mathematics curriculum. He seeks to place the programmable calculator in the historical, political, and sociological context of supplementary devices and teaching techniques. Then he relates some of his own experiences teaching with programmable calculators to suggest that they are here to stay but require appropriate curriculum development efforts to maximize their instructional value.

Krist presents several pedagogical roles that programmable calculators play in educational settings. She gives specific examples appropriate for the high school curriculum that provide insights into the interaction of students, calculators, and mathematics. Her examples include an open-ended discovery activity on logarithms; an exploration of a problem to find how many perfect squares there are among the numbers $X_1, X_2, X_3, \ldots X_{1000}$ (where for each n 1,2,3, ..., Xn = 9n + 7), and an interesting exercise to find when °F = °C?

Muser discusses an instructional unit with fourth- and fifth-grade students which focuses on learning more about problem solving. He describes several activities in which children actually used programmable calculators to count (by 2s, 3s, squares, etc.); to calculate sums of various integers (consecutive integers, consecutive even integers, consecutive odd integers, etc.); and, finally, to solve a problem requiring partial sums of the triangular numbers.

Huber presents, by example, a philosophical issue related to the use of programmable calculations appropriate for advanced high school students, college students, and teachers. He derives a formula for the ratio of the perimeter to the diameter of a regular inscribed polygon, shows that the formula is unstable for computing devices, and then modifies the formula to obtain a stable algorithm.

Snover's article describes a standard algebra II assignment, to plot the graph of $y = a \cdot (x + p)^3 + q$, and shows how the use of a programmable calculator enables students to advance beyond the simple graphing of a function to be able to grasp important generalizations. He focuses on the effects of varying the parameters a, p, q on the graph and gives an interesting game called "Catapult" to reinforce the ideas presented.

Johnson provides in her article an example from trigonometry of an approach to solving a problem that is impractical (because of the difficulty of calculations) without the use of calculating devices. She considers the problem of finding the third side and remaining angles of a triangle given two sides and the angle opposite one of them. Using the Law of Cosines rather than the Law of Sines, she proceeds to show how to find the unknowns avoiding the ambiguity which arises when there are two values of arcsin x (as possible angles of the triangle) for $0<X<1$.

Battista presents for enrichment a simulation game called "Ghostship," which can be used with high school students. In the article he suggests several questions, activities, and mathematical extensions which students might explore. These include discussing ideas universal to programming such as flow charts, loops, conditional tests, and branching; asking students how they would calculate the distance between the Ghostship and a missile shot if they knew the polar coordinates of both points; challenging students to develop a program to allow a second calculator to destroy the Ghostship; and exploring the game using rectangular rather than polar coordinates.

Haggerty illustrates the use of programmable calculators to run simulated experiments. He presents a probability experiment appropriate for high school students. Included in the article are a program and several interesting questions about the experiment to use in the classroom.

Mohler notes that programmable calculators can be quite useful in the mathematics classroom as a vehicle for stimulating independent study. He illustrates his point by presenting a discussion of Pythagorean triples. Flow charts and programming suggestions are presented for several problems given.

Weaver suggests a specific use of the programmable calculator (as a function machine) to show how one can direct attention to the underlying nature of mathematics, focus upon significant mathematical ideas, and exemplify an important type of learning and instruction. He illustrates his point by presenting four problem contexts appropriate for students in many school and teacher-preparation settings. In an end note, readers are invited to write for programs for a variety of machines.

Maor, in the context of discussing highlights from the history of π, gives different ways of computing π, including the methods of Archimedes, Viete, Wallis, Gregory, and Euler. He includes two programs and the article should be of interest to upper-grade high school students, college students of mathematics, and teachers.

Elich describes a one-quarter-credit programmable calculator techniques course with twelve class meetings given at Utah State University. The course is offered in two sections, one for algebraic calculators and the other for RPN calculators. The syllabus for each section is presented and twelve sample problems from the exercise sets are given.

Coup presents a bibliography of 145 programmable calculator resources including books, articles, ERIC documents, and dissertations. She lists the books and classifies the remaining references into seven categories: Bibliographies, Elementary School, Secondary School, College and Postgraduate, Games, Other Uses, and General. Many of the citations are annotated for the reader's benefit.

Mark A. Spikell
Editor

THE PROGRAMMABLE CALCULATOR: FAD OR SOLUTION?

Gerald R. Rising
Department of Instruction and Mathematics
State University of New York at Buffalo
Amherst, New York 14260

Once again we in the schools are confronted with something new for which extraordinary claims are mounted. This time it is the programmable hand-held calculator. But we need not have been around education for too long to have experienced similar episodes many times before. Only my age will make my list of innovative devices and programs longer than yours: the opaque projector, the tachistoscope, the slide projector, and the overhead projector; programmed instruction and computer-assisted instruction, individualization and levels programs; audio tapes, movies, television, and TV tapes; the computer and now the calculator.

In the schools today, however, I find more of these one-time panaceas on storage closet shelves or in file drawers than I do in operation in classrooms. For some of this I can only say, thank goodness; but in other cases I am convinced that a better response is sadness.

Before I turn to the programmable calculator, it may be instructive to consider the history of one of these earlier tools of education. To this purpose I offer a brief account of the rise and fall of programmed instruction. PI was one of the first educational products of the behaviorist psychologists who sought practical application of their powerful new theories. For those who did not pass through this educational phase of the 1960s, I note that PI replaces the textbook--and too often, as we will see, the teacher as well--with materials with which the student must constantly interact. Blanks are left to fill in (with the answer nearby for immediate feedback) and content is addressed in small steps that accumulate rapidly to give larger increments of learning. Although there are some technical differences among types (linear and branching, Egrule and Ruleg), the development is generally a (monotonous) series of tell and tell-back interactions.

It works. For a wide range of school learning and for many students, a great deal is learned through PI. The famous--perhaps infamous is a better word--Roanoke experiment displayed achievement

---

Gerald Rising is a former member of the Board of Directors of the National Council of Teachers of Mathematics. He is the author of numerous articles and books and recently completed work on a funded project to develop material for programmable calculator use in grades 11 and 12. His current research interests include instructional programs for gifted students and the development of secondary school curriculum materials.

results comparing school algebra classes of three types: (1) teacher taught, (2) PI taught with teacher support, and (3) PI taught with no teacher present. High to low rank in achievement for these groups was: (3), (2), and (1).

This tremendous success was based on some high quality PI books developed by Jack Forbes and others and published by Encyclopedia Britannica Press. These materials and those that followed briefly swept the country. In thousands of mathematics classrooms[1] student-teacher interaction was replaced by silent, individual PI workbook activity.

But where are the PI materials today? Try to find a set. In a few curriculum libraries, possibly; in school classrooms, never. Why? Because the all-too-common neutralizing factors came into operation: entrepreneurial oversell, quality dilution as the second and third raters entered the field, overmechanization, and in the classroom overuse and misuse. As just one example of the kind of overclaim that turned many away from these materials, I recall a speaker at an NCTM national meeting announcing that the 300 percent acceleration in learning rates produced by PI would mean that students would soon regularly achieve college graduation at age twelve to fourteen. The predictable reaction set in and PI retreated first to the storeroom and then to the incinerator.

Too bad. I am one of those who reacted most strongly to the "this will solve everything" aspect of PI promotion, but I believe that we did indeed let out the baby with the bathwater. Programmed instruction used judiciously could (and too rarely today does) support instruction. For example, it could provide an excellent device to help a student absent for a protracted period. In my university's Chemistry Department it is used, in its alternative form CAI[2], to provide short instructional sequences; for example, teaching how to perform Ph calculations. In general, though, this significant support tool is lost to instruction.

So now we have a new toy to consider, the programmable hand-held calculator. Will it follow its ancestors to the storage cabinet? It is difficult to predict anything else since its immediate predecessor, the pocket calculator, is already there in many schools.

---

[1] PI addressed other subjects as well but as usual math got hit first and hardest. Why? Because math is seen by nonmathematicians and by too many mathematicians as a rote subject--essentially computation--to be accomplished by drill.

[2] For Computer Assisted (or Aided) Instruction.

Still, there are possibilities here. What saves the pocket calculator from extinction will probably also save the programmable calculator--what I call the Christmas market. The ubiquitous calculator--more sold each year than TV Guide, the most widely distributed magazine--appears to solve the present purchasing penchant for just about everyone in this country. It works both ways: "He's interested in math so I'll get him a calculator," and "He can't do math and hates it so I'll get him a calculator." Extension or replacement, this low-cost device seems to hold a powerful attraction our psyches just cannot resist.

At any rate this provides us in education with a fortuitous availability, and the mass market also depresses prices. It is difficult to imagine the extraordinary depth of this last effect when combined with rapid technological development. Programmable calculators today, some (most notably the TI-57) with price tags under fifty dollars, offer calculation power roughly equivalent to a computer costing over $500,000 just 25 years ago. If my data are accurate (and you need only adjust the date if I fall short), this represents a reduction factor of 10,000, and that in the face of serious inflation of other prices. Who says that the space race never gave us anything?

All well and good: they'll be around, it seems. Can we in the schools make use of them? The answer to that question is a simple and direct, Yes! And an equally simple and direct, No! I am not waffling. The answer is completely situation-dependent. In about 90 percent of the situations into which a programmable calculator is inserted today, the result is pedagogical disaster. In the other 10 percent they produce a resounding success.

First the bad news. The settings that won't work are quite predictable: the principal orders 30 calculators to fill out his purchasing budget so he won't be cut next year. The teacher buys a set "to motivate the kids." Another buys them to check answers, and still another to respond to lack of basic skills. What these examples have in common is what distinguishes them from those settings in which calculators can contribute effectively. They are ad hoc solutions thoughtlessly offered for what are usually inappropriate problems. They are unplanned. And most important they pit only this little piece of electrical equipment against the mighty monsters of today's classrooms. Somehow this little black box--on its own--is expected to perform eminently human instructional tasks. It is to their credit that they last even a few minutes before students begin to test their tensile strength in drops from increasing heights.

Surely there is a better way. Despite my reservations about ill-designed settings, I believe that the programmable hand-held calculator has an honorable and a distinctive role to play in the school instructional program. As a tool it can help students to gain deeper insights into serious mathematical ideas--function and real number, for example--and it can introduce major calculation efficiencies into the teaching-learning program. But it cannot do any of these things without the absolutely necessary attendant pedagogical softwear: both textual and teacher support materials.

I say this based not only on my historical observations, but also on my first-hand experience in the classroom. Teaching about and with calculators is different. I went into the classroom thinking at first only of the opportunities for individual experiment and creative activity. Indeed I was able to carry off some of these opportunities. But I soon came to realize that there is also much instruction at the opposite end of the creative-rote spectrum. It is often necessary to impose the strictest "push this key, then this key, then this key" regimen in order to communicate specific technique.

My colleagues, Betty Krist, Carl Roesch, and Don Stover, and I are convinced that our students learned many of the concepts (in this case of eleventh- and twelfth-grade mathematics) with greater understanding when they were taught by the techniques and with the curricular materials we developed for their use with programmable calculators. We were able to give them more efficient access to ideas: by programming a function, for example, they were able to plot its graph quickly and accurately, thus getting directly to the regularities that graphing exposes. We were able to provide creative experiences in the development of traditional content: for example, the students on their own discovered the role of the LOG key, thus efficiently introducing this important topic in a meaningful way.

We were also able, we feel, to communicate new ideas effectively. Most important of these are programming concepts. The central ideas of programming, its power in particular, are often hidden or disguised in computer science courses by the complexities of programming languages. Here the very simplicity of "giving the R/S key a designated role" is uncluttered and clear. The other keys to programming power, branching and decision making, are also so simplified and straightforward that they are also almost impossible to misunderstand. And once they have this basic understanding students are far better prepared, if they wish to do so, to make their way through the forest of computer science languages.

I only wish that every serious teacher could have the opportunity that my colleagues and I had using these tools with appropriate support materials in the classroom. I am certain that they would be sold on programmables. Given this kind of pedagogical software support, some of which will be provided in the papers in this collection, I am convinced that any teacher of reasonable quality would have his or her instruction powerfully enhanced by these devices.

Let us then get to our educational task. Let us provide the support necessary to allow this important tool to take its rightful place in our instructional programs. The designation of the hand-held programmable calculator as an electronic slide rule is apt if inadequate. Like the slide rule should have been, just so the programmable should be incorporated at every level of instruction possible. But to do so adequately we must first do our pedagogical homework.

CALCULATOR PEDAGOGY

Betty J. Krist
West Seneca East Senior High School
West Seneca, New York 14224

During the past two years we have been working with two classes of students in two different schools who are studying eleventh- and twelfth-grade mathematics with the aid of programmable calculators. This project, supported by the National Institute of Education Basic Skills Group grant #400-78-0013 and entitled "Grade 11-12 Curriculum Modification Reflecting the New Computation," was designed to develop, test in classrooms, and revise into final form curricular materials supporting programmable calculator usage in eleventh- and twelfth-grade mathematics. The project staff, Gerald R. Rising, Betty J. Krist, Carl J. Roesch, and Donald W. Stover has produced eight chapters of eleventh-grade text and seven chapters of twelfth-grade text. These materials were used with students at Sweet Home Senior High School, Amherst, New York and West Seneca East Senior High School, West Seneca, New York. The students were each issued a Hewlett-Packard model 33E or model 25 scientific programmable calculator. These calculator models are almost identical, using Reverse Polish Notation logic and having 50 program steps.

The eleventh-grade text content was based on the New York State Regents Eleventh Year Mathematics Curriculum, but was modified to reflect the technological power available to these students. For example, logarithms were studied as a function rather than simply as a computational tool and such time-honored topics as interpolation were bypassed in favor of functional analysis. The students were specifically instructed in the use of the calculator. For roughly one-half of the time the course was taught without use of the calculator, because the content of that part of the course was not considered to be enhanced by calculator usage. The twelfth-grade material was developed for a one-semester course based on standard twelfth-grade topics that could benefit from calculator availability.

No Significant Difference: A Type II Error at Least

The students in these classes were randomly chosen from volunteers. They were students who had completed the traditional ninth- and tenth-grade Regents program. Their prior college preparatory mathematics achievement ranged from very poor (barely passing Math

---

Betty Krist teaches grades 10 and 12; her twelfth graders use programmable calculators. She is also co-director of the Gifted Math Program at SUNY-Buffalo, a program to provide modern mathematics to highly verbal, well-motivated students in the upper 10-15 percent of the school population.

10) to outstanding. They had the interest, initiative, and parental support to volunteer for an experimental program that involved calculators. None of them had had any experience with either scientific or programmable calculators.

These students took two final examinations in Math 11, the standard Regents examination and a final examination prepared by the project staff. When the students took the Regents examination, they were not allowed to use calculators; when they took the project examination, they used the same calculators they had been using in their work throughout the course. An analysis of variance of their test scores does not refute the null hypothesis that there is no significant difference between allowing students to work with calculators or without them. We note here the necessary reservation that there may have been conceptual differences in the tests as well.

Some caution must be exercised in viewing this analysis because this was not intended to be a statistical study and the standard controls were not specifically mounted. However, the preceding analysis of the students' work is much like the typical style of empirical research that generally has been conducted about calculators. It represents the place where most studies end and where our studies seem to have begun. We contend that this analysis is exactly a "type II error" in its implications: we accept a false null hypothesis of no difference between groups. This "no difference" hides what we are interested in: the pedagogical roles that the programmable calculator can play in an educational setting.

In working with senior high students who used programmable calculators as an aid in their study of mathematics, we were not only concerned with materials but with the interaction of students, calculators, and mathematics. We looked carefully at many aspects of instruction and desired to expose some of the gains to be achieved in using calculators and also some of the losses. While discovery, creativity, and problem-solving were particular aspects of pedagogy that were carefully scrutinized, we also viewed the calculator not merely as a computational aid but anthromorphically as a communication device. The calculator, it turns out, is a pedagogical language between student and teacher and between student and self that can reveal some of the meta-level effects of information processing and student thinking about mathematics.

We turn now to some specific examples that provide insights into the interaction of students, calculators, and mathematics.

Logarithms

This lesson has been used by three different teachers (Rising, Roesch, and Krist) with three different groups of students. In each case the results have been dramatically similar. This is also an example of an open-ended discovery activity with an entire class of eleventh-grade mathematics students who were currently studying exponents. Here are notes from one class.

The class began with a very simple direction: "Your calculator has a key labeled [log]. What does this key do?" One student was chosen to be secretary to write notes on the board. The teacher merely took a seat in the back of the room. The students began by making the following table:

n	log n	n	log n
0	error	5	.7
1	0	9	.9542
2	.3010	6	.7782
8	.9031	7	.8451
3	.4771	-1	error
4	.6021	100	2
10	1	1000	3
		10000	4

At this point two students made the following comments: "This is not the real thing--it's rounded--3 should be .477121255, but even that's rounded." "This is getting us nowhere, we should look for some patterns." With this, the teacher suggested that students make a list of conjectures and look for relationships within the table. At this point, the students began to work independently, making the following list of conjectures:

1) $10^{\log n} = n$, $10^{.6021} = 4.0004$

$10^{.3010} = 1.9999$, $10^1 = 10$, $10^2 = 100$

(This rather startling *initial* conjecture was the first to be expressed in *each* of the three classes where this lesson was used.)

2) 5, 50 and 500 have the same decimal (in the log) and the same number in front of the zeros.

3) Rule 2 also works for 1, 10, 1000;
3, 30, 300;
etc.

4) But .5 doesn't work like the 5s.

5) But .2 extends the 5s series .2 → -.6990, so there is a relation between 2s and 5s.

6) Negative logs give (the number of) zeros in front.

.5 → -.3010
.05 → -1.3010
.005 → -2.3010

og .5 + log .2 = 0

og 2 + log .5 = 0

hat's the relation between 2s and 5s.

$\log n + \log \frac{1}{n} = 0$

og 5 - log .5 = 1

og 2 - log .2 = 1

og n - log .n = 1*

|log n| + |log .n| = 1

he number before (the log) is like the exponent in scie
ific notation. Another student added, "That's O.K., bu
ot for negatives--it's one less than the number of digi

lass discussion stopped and the teacher made the follow
'Consider the following table:

n	log n
1	0
2	.3010
3	.4771
4	.6021
5	.6990
6	.7782
7	.8451
8	.9031
9	.9542
10	1.0000

ny relations among the numbers on this table?"

nt comments again ignited:

3 give 6: .3010 + .4771 = .7782 almost

4 give 8: .3010 + .6021 = .9030 almost

3 give 9 exactly: .4771 + .4771 = .9542

asked students to complete the following table as well
ld without using their calculators.

n	log n
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	

ents made the following entries:

n	log n	
11	1.0396	halfway between 10 and 12
12	1.0792	3 and 4 or 2 and 6
13	1.1127	halfway between 12 and 14
14	1.1461	2 and 7
15	1.1761	3 and 5
16	1.2041	2 and 8 or 4 and 4
17	1.2297	halfway between 16 and 18
18	1.2552	2 and 9 or
	1.2553	3 and 6
19	1.2782	halfway between 18 and 20
20	1.3010	10 and 2 or
	1.3011	4 and 5

er asked the students to check the logs by using their rs. The new table contained discrepancies with the old

n	log n
11	1.0414
12	1.0792
13	1.1139
14	1.1461
15	1.1761
16	1.2041
17	1.2304
18	1.2553
19	1.2788
20	1.3010

the class period was drawing to a close, so the teacher homework: "Look at our tables and conjectures and try to rs or decide why our statements are true."

The next day's class opened with the following pronouncement fr one student: "The key is the first statement: $10^{\log n} = n$. The res is just how exponents work. The errors are because of rounding." Wh asked about the logs of 11, 13, 17, and 19, he said, "Exponents don't have even jumps like $2^2 = 4$, $2^3 = 8$, $2^4 = 16$. The differences are no the same." With that the entire class sat back. Their discussion ha ended and they were satisfied with their results. The remainder of this and subsequent classes contained the usual theorems about logar- ithms and the students worked with bases 2, 3, 2.3, 3.1, and e. They did not do computations with logs but rather considered logs as a cla of functions and studied them, as a collection, from that viewpoint.

This episode has much dramatic significance. First, it shows that genuine discovery activities are exciting. These students were enthusiastic about their work. They seemed to be working on a detective story. They enjoyed the clues given by their calculator. No one said, "Just tell us how it works." It would seem that they sensed that the teacher would have spoiled their game if she had immediately answered all their conjectures as they arose. These students obtained the important properties about logs and why they were true, by themselves. They even discovered a few rather obscure properties of logs, e.g., $|\log n| + |\log \frac{n}{10}| = 1$ $(1 \leq n \leq 10)$. Their statement that $10^{\log n} = n$ was generalized to $b^{\log_b n}$ and became the basis of all the subsequent proofs and classwork.

Secondly, the calculator was a device that provided a framework for the activity and helped the students put their own ideas together. This activity is hard to image in a classroom with log tables rather than calculators. The calculators provided a setting and they were an aid to student thinking, but the students had to go beyond the numbers that were exhibited in the calculator's display.

Problem 513

In examining the work of students solving problems, we can observe another example of the pedagogical language aspect that a calculator can bring to our classrooms. In this example the calculator is not an integral part of the activity or even the solution, but it did aid the students in reaching their solution. The students considered:

"Problem 513: For each $n = 1, 2, 3, \ldots$, let $x_n = 9n + 7$. How many perfect squares are there among the numbers $x_1, x_2, x_3, \ldots$ 1000?"* Figure 1 contains one particular student's solution. He searched for patterns and found several.

Scott Mayer

9025 $\quad$ 9.25

For each n = 1, 2, 3, ..., let $x_n = 9n + 7$. How many perfect squares are there among $x_1, x_2, x_3, \ldots, x_{1000}$?

no neg.'s because can't have $\sqrt{-x}$

5 9 0 1
Enter
9
x
7
+
F $\sqrt{x}$
9000

4 → x = 16
5 → x = 25
13 → x = 169
14 → x = 196
22 → x = 484
23 → x = 529
31 → x = 961
32 → x = 1024
40 → x = 1600
41 → x = 1681
49 → x = 2401
50
58
59
67
68
76
77
85
86

7 x's

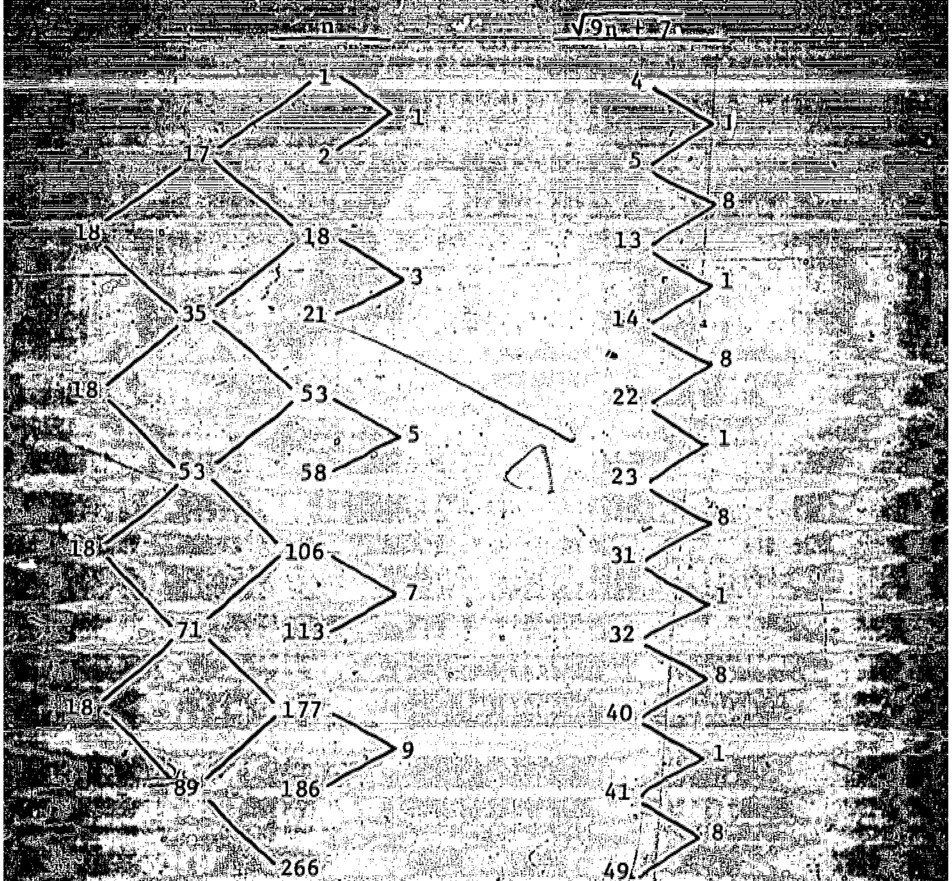
$x_{1000} = 9000 + 7$
$= 9007$

21

981 – 94 – x = 8,836
002 – 95 – x = 9025

$\sqrt{9{,}007} = 94.9052$

103
104
110.3



elieving that these patterns should continue he announced his olution, "There are 21." When asked if he was sure, he said onfidently, "Yes," and the other students quit working to listen o his explanation. The teacher asked for a proof. Another student aid, "You could write a program." He proceeded to furnish the ollowing program:

01	1	11	g x=0
02	STO 1	12	GTO 16
03	RCL 1	13	1
04	9	14	STO + 1
05	X	15	GTO 03
06	7	16	RCL 1
07	+	17	R/S
08	f √x	18	RCL 2
09	STO 2	19	R/S
10	g FRAC	20	GTO 13

This program takes approximately 19 minutes to run (on an HP-33E) from 1 to 1000. It produces exactly those numbers that the first student had predicted. The program does not use any of the first student's patterns. It evaluates each of the numbers x_1 to 1000 and tests for perfect squares.

There is an interesting comparison of logic vs. calculation here. The first student quickly generated examples using his calculator but he abandoned the calculator in favor of pure thinking. The calculator helped him gather his ideas together, but it was not the forum in which he chose to solve the problem. Once he put his ideas together it was quicker for him to work without the calculator.

A brief third example can balance the overwhelming positive aspects of these first two examples and proclaim that the calculator is not the total answer to our problems.

When Does °F = °C?

When these students were just beginning their work with calculators they were presented with the following exercise:

Using one of the two conversion formulås for Celsius and Fahrenheit temperatures $C = \frac{5}{9}(F-32)$, $F = \frac{9}{5}C+32$, answer the following:

1. C = 100°, find F.
2. Convert 32°F to C.
3. Change 68°F to C.
4. Change 98.6°F to C.
5. Find by experimenting when F and C are the same.

The students had no trouble doing 1 through 4, but few could answer 5. A typical response was, "I don't know how to do it."

A few students did try a few specific values for F and C, but their choices were erratic and their results were not organized in any way. One student said, "If F < 32 then C is negative," when he had noticed that as F decreased F − C decreased, but he rejected the idea that F and C could be identical and negative. Only three students correctly solved the problem on their own.

What these three classroom examples clearly illustrate, among other things, is that a calculator can be an important aid to student thinking about mathematics and a pedagogical language. A critical word of that sentence is can. What we need to consider are appropriate settings to allow this to happen and to recognize that it possibly will not happen. What each of these examples required was some additional thinking from students. The calculator was helpful to initiate student thinking, but the students needed to go beyond the machine. This is precisely what we want our students to do: be capable of thinking about mathematics for themselves.

USING PROGRAMMABLE CALCULATORS TO ENLARGE THE PROBLEM-SOLVING WORLD OF 10-12 YEAR OLDS

Gary L. Musser
Oregon State University
Corvallis, Oregon 97331

Calculators and microcomputers are finding increasing acceptance n school programs. Although many teachers and parents still believe hat calculators will provide us with a generation of mindless (and athless) children, studies have shown the contrary. Calculators can e used to promote the learning of traditional arithmetic as well as rovide opportunities for many more problem-solving experiences. Micro omputers, when backed up with effective courseware, can be a remendous aid to individualizing instruction. In addition, micro-omputers can be used to teach children programming skills to enable hem to solve complicated problems.

Even though microcomputers are becoming more affordable, few if any) schools can provide one for each child (or even each pair f children). Happily, programmable calculators currently fill the ap between calculators and microcomputers both in price and apability. For less than $100 one can purchase a high quality rogrammable calculator which can be used to introduce children to olving problems via programming. Moreover, an entire classroom set for use in one school building, say) can be purchased for the price f two or three microcomputer systems. Thus, if one of your goals s to promote problem solving through programming, the potential ayoff of each child being able to work with his or her own program-able calculator in school is enormous. Since the technological gap etween programmable calculators and computers is narrowing, they ill likely become synonymous within 20 years (hopefully much sooner) nd then each child will have his or her own microcomputer.

In September, 1979, I was asked to work with 13 talented fourth-nd fifth-grade students who were involved in a mathematics enrichment rogram for two 35-minute periods per week. Because I had access to 3 HP-33E programmable calculators, this situation presented a unique pportunity to witness firsthand the students' capabilities in using rogrammables, and to judge their enthusiasm for writing programs to olve problems. As I had suspected, the students immediately fell in ove with the calculators. In fact, during the seven weeks I worked ith them, it was difficult to hold prolonged discussions because they lways wanted to be working with the calculators.

Gary Musser teaches preservice and inservice elementary and econdary teachers. He has given many talks and workshops on a variety topics and several of his articles have been published in such urnals as the Mathematics Teacher and the Arithmetic Teacher.

Originally presented at the Symposium, this article was subsequent blished in The Computing Teacher, Vol. 8, No. 1, September 1980.

The main goal of our instructional unit was to learn more about problem solving, particularly how to apply techniques of programming to solve problems which would not be within reach of a typical mathematics program at this level. A second goal was to help the children see that certain problems had various solution types and that sometimes a programmed solution might be "best," whereas an insightful mathematical solution might be "best" other times. The following discussion provides a glimpse into the students' accomplishments during our seven-week session.

After becoming familiar with many of the keys of the calculator, we constructed our first program. (Our calculators used Reverse Polish Notation, which is a great convenience and posed no difficulty to the children.) We programmed the calculator to count by 1s as follows:

a. The number 1 was entered in Storage Register 0.

b. The following program was entered:

Step	Operation	
01	RCL 0	This step recalls a 1 from Register 0.
02	+	This adds the 1 to whatever we have.
03	f PAUSE	This makes the calculator pause for 1 second so we can see the number.
04	GTO 01	This sends us back to Step 01.

When the children pressed their Run/Stop keys, they were thrilled to see their calculator count. It was easy for them to get the calculator to count by 2s or 3s or any other digit by simply changing the 1 in Register 0 to 2 or 3, etc.

Our next step was to try to count using the odd numbers. Once the students observed that each odd number was one less than an even number, they wrote the following program.

a. Store the number 2 in Register 0.

b.	01	RCL 0	
	02	+	This counts by 2s and stores the results
	03	STO 1	in Register 1.
	04	1	This produces an odd number.
	05	-	
	06	f PAUSE	This permits the calculator to display the odd number.
	07	RCL 1	This recalls our last even number and
	08	GTO 01	returns to the beginning.

After the students had run this program successfully, I listed the following two programs on the board and asked them to determine the output for each program.

a.	Store the number 1 in Register 0.		a.	Store the number 2 in Register 0.	
b.	01	RCL 0	b.	01	1
	02	2		02	fPAUSE
	03	X		03	RCL 0
	04	1		04	+
	05	−		05	GTO 02
	06	fPAUSE			
	07	1			
	08	RCL 0			
	09	+			
	10	STO 0			
	11	GTO 01			

To obtain the outputs, the students pretended that they were the calculator. Each in turn executed one step of program just as the calculator would. For example, for the program on the left above, they would say "1, 2, 1 x 2, 1, 1 x 2 - 1, Pause, 1, 1, 2, Store in 0, Go to 1, etc." To their amazement, the outputs of all three programs were identical. This gave us an opportunity to observe that programs (and problems) can have different solutions and that many times it is worthwhile to try to write programs more simply and, perhaps, with fewer steps. After counting by the odd numbers, they wrote a program to count by the squares; i.e., 1, 4, 9, 16, etc.

Our calculators had an easy method to accumulate numbers as we counted them; namely, the command "STO + 1" would "'store' the number currently displayed by adding it to register 1." I asked the students to write a program which would count by odd numbers (1,3,5,7,9,...) and add them as it went along (1, 1+3, 1+3+5, ...). After a considerable amount of work, we arrived at a program that produced the desired sums. After several of the sums were displayed, one student exclaimed, "Hey, this program is giving us all the square numbers." Upon further examination the student could actually see why the sums of consecutive odd numbers beginning with 1 always produced square numbers by considering the following arrays:

1   1 + 3   1 + 3 + 5   etc.

The next day I told the famous story about Gauss having to add the numbers from 1 to 100. When asked how they would work this problem, the students separated into two groups; about half wanted to use pencil and paper, and the rest wanted to write a program. Before the end of our period, all of the calculator students had written and run their program and none of the paper-and-pencil group had come close to the correct answer.

At our next meeting I asked the students how they would add the numbers 1 to 1000. They all said, "By using a programmable calculator." Then I mused, "How long will it take if we have one fPAUSE in our program?" Since an fPAUSE lasts about one second, their reply was, "About 1000 seconds." Since we did not have time to run such a program, I showed them how Gauss was supposed to have done the 1 to 100 problem:

$$\begin{array}{rrrrrrr} 1 & + \; 2 & + \cdots & + \; 99 & + \; 100 \\ 100 & + \; 99 & + \cdots & + \; 99 & + \; 1 \\ \hline 101 & + \; 101 & + \cdots & + \; 101 & + \; 101 \end{array}$$

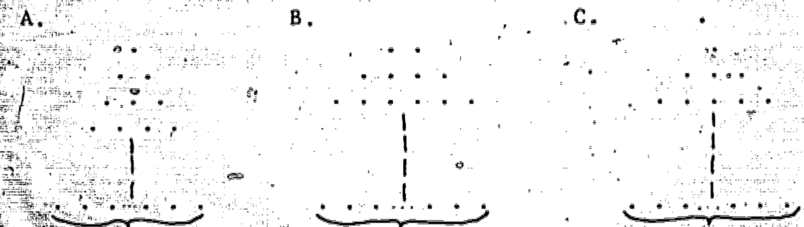
Therefore, two $1 + 2 + \cdots + 99 + 100$ equals 100×101,
so $1 + 2 + \cdots + 99 + 100 = \frac{100 \times 101}{2} = 5{,}050$.

After they had calculated $1 + 2 + \cdots + 1000$ in a similar fashion using their calculators for the multiplication and division, we discussed how problems involving calculations could be done in several ways: mentally, using paper and pencil, using a calculator, using a programmable calculator, or any combination of these with "mathematical reasoning" as Gauss did.

Since Christmas was nearing, I presented the following problems to solve:

I. The Christmas trees below are composed of ornaments and are to appear on greeting cards (i.e., 2-dimensional).

A. B. C.



200 ornaments in the last row — 200 ornaments in the last row — 199 ornaments in the last row

How many ornaments are in each tree?

All students recognized tree A as a Gauss-type of problem, which they solved much as Gauss did. The only solution obtained for problem B was a program which summed the even numbers. Most students recognized problem C as the sum of odds, which immediately led to finding the appropriate square (100^2, in this case) as we had done earlier. After problem C had been solved, I asked the students if they could see any connection among the three problems. After a few hints they observed that the number of ornaments on tree B together with the number of ornaments on tree C total the number on tree A. Thus, if problems A and C were solved first, the solution to problem B could be obtained using subtraction.

II. Now imagine the two 3-dimensional trees made from spherical ornaments. Tree D has 1 ornament on the top layer, 3 on the second, 6 on the third, 10 on the fourth, etc. (these are the triangular numbers), down to the bottom layer which is an equilateral triangle with 100 ornaments on each side. Tree E has the same configuration except the layers are represented by the square numbers $(1,4,9,16,\ldots,100^2)$. How many ornaments are required to make these trees?

One student recognized the numbers in each layer as the partial sums leading up to Gauss' problem. Having successfully written a program to find "Gauss" sums, it was an easy step to write a program for tree D:

01	RCL 0	
02	1	This counts by 1
03	+	and stores the numbers in
04	STO 0	Register 0, and
05	fPAUSE	pauses to show the number of the layer.
06	STO + 1	This adds consecutive numbers to Register 1
07	RCL 1	forming "Gauss" sums and recalls these sums.
08	STO + 2	
09	RCL 2	This adds successive "Gauss" sums and pauses
10	fPAUSE	to display them.
11	GTO 01	

The answer to this problem is 171,700. The solution to the Tree E problem can be found in a similar manner.

These final problems involving Christmas trees were extremely useul in helping the children to see the role of a programmable calculator is a tool in the problem-solving process. Where the problems involving rees A, B, and C were most easily solved using mathematical thinking Gauss' Method) and patterns (the sums of consecutive odd numbers ncluding 1 are square numbers), the only techniques available to olve the fourth and fifth problems were programs for the calculator. roblems like these, which formerly were beyond the realm of even high chool students, can now be solved fairly routinely by some fifth-grade tudents.

This unit I taught fifth- and sixth-grade students would also be appropriate for students in grades 8 and up. (I am going to integrate it into a class for elementary teachers I teach.) Judging from the excitement which came out of this work with programmable calculators, teachers who can incorporate teaching problem solving via programmable calculators will have an appreciative audience: students, parents, and administrators, alike.

STABILIZING ARCHIMEDES' ALGORITHM FOR PI

John Huber
Department of Mathematics
Pan American University
Edinburg, Texas 78539

Using the fact that the circumference of a circle lies between the perimeter of any regular inscribed polygon and that of any regular circumscribed polygon, Archimedes was able to show that π lies between 223/71 and 22/7. The purpose of this paper is to derive the usual recursive formula for the ratio of the perimeter to the diameter of a regular inscribed polygon, show that this formula is unstable for computing devices, and modify the formula to a stable algorithm.

Let s_n denote the length of a side of a regular polygon of n sides inscribed in a circle of radius r, and let s_{2n} denote the length of the side of the regular polygon of 2n sides formed by bisecting the arc containing consecutive vertices of the original regular inscribed polygon of n sides.

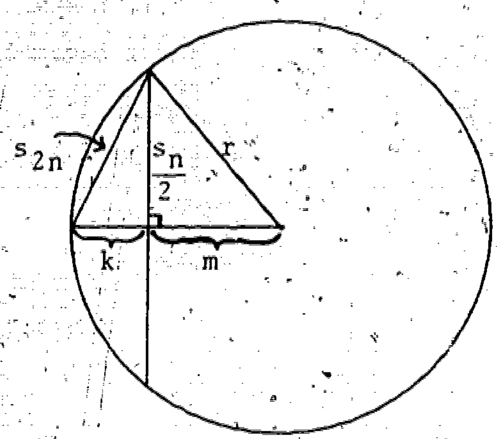


Figure 1

John Huber teaches mathematics education courses for K-12 teachers. He has a wide range of interests including research on mathematics attitude and anxiety, cognitive processes in learning algebra, and the use of calculators and computers in secondary mathematics.

Using Figure 1 and the Pythagorean Theorem we have:

$$r^2 = \left(\frac{s_n}{2}\right)^2 + m^2 \qquad (1)$$

and

$$m = \sqrt{r^2 - \frac{s_n^2}{4}} \qquad (2)$$

Since $k + m = r$, we have:

$$k = r - m \qquad (3)$$

and substituting (2) into (3) gives:

$$k = r - \sqrt{r^2 - \frac{s_n^2}{4}} \qquad (4)$$

so

$$k^2 = r^2 - 2r\sqrt{r^2 - \frac{s_n^2}{4}} + r^2 - \frac{s_n^2}{4} \qquad (5)$$

$$k^2 = 2r^2 - 2r\sqrt{r^2 - \frac{s_n^2}{4}} - \frac{s_n^2}{4} \qquad (6)$$

Again using Figure 1 and the Pythagorean Theorem, we have

$$s_{2n}^2 = k^2 + \left(\frac{s_n^2}{2}\right)^2 \qquad (7)$$

and substituting (6) into (7) gives:

$$s_{2n}^2 = 2r^2 - 2r\sqrt{r^2 - \frac{s_n^2}{4}} \qquad (8)$$

$$s_{2n} = \sqrt{2r^2 - 2r\sqrt{\frac{4r^2 - s_n^2}{4}}} \qquad (9)$$

$$s_{2n} = \sqrt{2r^2 - r\sqrt{4r^2 - s_n^2}} \qquad (10)$$

Then considering a regular hexagon inscribed in a circle of radius 1, (10) becomes:

$$s_{2n} = \sqrt{2 - \sqrt{4 - s_n^2}} \qquad (11)$$

where $s_6 = 1$ and perimeter/diameter $= \frac{n \cdot s_n}{2}$. Using a programmable calculator, we have the results in Table I. (See Appendix for programs.) Clearly the ratio does not converge to π.

Knowing that $\lim_{n \to \infty} \frac{n \cdot s}{2} = \pi$, why does the algorithm not converge on the calculator? The lack of convergence is caused by the large relative error in the difference $2 - \sqrt{4 - s_n^2}$. Since $\sqrt{4 - s_n^2}$ is close to 2 (see Table II), the rounding error along with the closeness of 2 and $\sqrt{4 - s_n^2}$ causes a large relative error in $2 - \sqrt{4 - s_n^2}$, resulting in an unstable algorithm (Conte and de Boor, 1972, pp. 13-14).

To stabilize the algorithm, we must remove the difference $2 - \sqrt{4 - s_n^2}$. This can be accomplished by rationalizing the numerator under the radical in (11), giving us:

$$s_{2n} = \sqrt{(2 - \sqrt{4 - s_n^2})\frac{(2 + \sqrt{4 - s_n^2})}{(2 + \sqrt{4 - s_n^2})}} \qquad (12)$$

resulting in:

$$s_{2n} = \sqrt{\frac{s_n^2}{2 + \sqrt{4 - s_n^2}}} \qquad (13)$$

Eliminating the difference results in a stable algorithm that converges to π. Using a programmable calculator, we have the results in Table III.

Reference

Conte, S. D. and de Boor, Carl. Elementary Numerical Analysis (2nd ed.). New York: McGraw-Hill, 1972.

TABLE I

Number of Sides	Length of Side	Perimeter/Diameter
6	1.0000000000	3.000000000
12	0.5176380902	3.105828541
24	0.2610523844	3.132638613
48	0.1308062585	3.139350203
96	0.0654381654	3.141031951
192	0.0327234633	3.141452473
384	0.9163622792	3.141557615
768	0.0081812081	3.141583911
1536	0.0040906127	3.141590529
3072	0.0020453076	3.141592407
6144	0.0010226544	3.141595284
12288	0.0005113277	3.141597288
24576	0.0002556658	3.141621319
49152	0.0001278348	3.141693413
98304	0.0000639218	3.141885657
196608	0.0000319687	3.142654499
393216	0.0000160000	3.145728000
786432	0.0000080623	3.170208743
1572864	0.0000041232	3.242542203
3145728	0.0000022361	3.517030823
6291456	0.0000014142	4.448731201

TABLE II

Number of Sides	$\sqrt{4 - s_n^2}$	$2 - \sqrt{4 - s_n^2}$
6	1.732050809	0.2679491924
12	1.931851653	0.0681483474
24	1.982889723	0.0171102772
48	1.995717846	0.0042821535
96	1.998929175	0.0010708250
192	1.999732276	0.0002677243
384	1.999933068	0.0000669322
768	1.999983267	0.0000167331

TABLE III

Number of Sides	Length of Side	Perimeter/Diameter
6	1.0000000000	3.000000000
12	0.5176380902	3.105828541
24	0.2610523855	3.132628613
48	0.1308062585	3.139350203
96	0.0654381656	3.141031951
192	0.0327324633	3.141452472
384	0.0163622792	3.141557608
768	0.0081812081	3.141583892
1536	0.0040906126	3.141590463
3072	0.0020453074	3.141592106
6144	0.0010226538	3.141592517
12288	0.0005113269	3.141592619
24576	0.0002556635	3.141592645
49152	0.0001278317	3.141592651
98304	0.0000639159	3.141592653
196608	0.0000319579	3.141592653
393216	0.0000159790	3.141592654
786432	0.0000079895	3.141592654
1572864	0.0000039947	3.141592654
3145728	0.0000019974	3.141592654
6291456	0.0000009987	3.141592654

APPENDIX

PROGRAMS FOR GENERATING n, s_n, and $\frac{n \cdot s_n}{2}$ using $s_{2n} = \sqrt{2 - \sqrt{4 - s_n^2}}$.

TI 58 and 59	
	LRN
00	2nd CP
01	STO
02	01
03	R/S
04	STO
05	02
06	R/S
07	x^2
08	RCL
09	02
10	+/−
11	+
12	4
13	=
14	$\sqrt{x}$
15	+/−
16	+
17	2
18	=
19	STO
20	02
21	2
22	2nd
23	Prd
24	01
25	RCL
26	01
27	R/S
28	RCL
29	02
30	$\sqrt{x}$
31	R/S
32	x
33	RCL
34	01
35	÷
36	2
37	=
38	R/S
39	GTO 08
	LRN
	RST

HP 33E	
	PRGM
00	f Clear Prgm
01	f FIX 9
02	STO 1
03	R/S
04	STO 2
05	R/S
06	g x^2
07	RCL 2
08	CHS
09	ENTER
10	4
11	+
12	f $\sqrt{x}$
13	ENTER
14	CHS
15	2
16	+
17	STO 2
18	2
19	STO x 1
20	RCL 1
21	R/S
22	RCL 2
23	f $\sqrt{x}$
24	R/S
25	ENTER
26	RCL 1
27	x
28	ENTER
29	2
30	÷
31	R/S
32	GTO 07
	RUN
	g RTN

INPUT	
6	R/S
1	R/S

OUTPUT	
R/S	n
R/S	s_n
R/S	Perimeter/Diameter

Appendix (continued)

PROGRAMS FOR GENERATING n, s_n, and $\frac{n \cdot s_n}{2}$ using $s_{2n} = \sqrt{\frac{s_n^2}{2 + \sqrt{4 - s_n^2}}}$

TI 58 and 59	
	LRN
00	2nd CP
01	STO
02	01
03	R/S
04	STO
05	02
06	R/S
07	x^2
08	RCL
09	02
10	+/-
11	+
12	4
13	=
14	$\sqrt{x}$
15	+
16	2
17	=
18	1/x
19	x
20	RCL
21	02
22	=
23	STO
24	02
25	2
26	2nd Prd
27	01
28	RCL
29	01
30	R/S
31	RCL
32	02
33	$\sqrt{x}$
34	R/S
35	x
36	RCL
37	01
38	÷
39	2
40	=
41	R/S
42	GTO 08
	LRN
	RST

HP 33E	
	PRGM
00	f Clear Prgm
01	f FIX 9
02	STO 1
03	R/S
04	STO 2
05	R/S
06	g x^2
07	RCL 2
08	CHS
09	ENTER
10	4
11	+
12	f$\sqrt{x}$
13	ENTER
14	2
15	+
16	g 1/x
17	ENTER
18	RCL 2
19	x
20	STO 2
21	2
22	STOx1
23	RCL 1
24	R/S
25	RCL 2
26	f$\sqrt{x}$
27	R/S
28	ENTER
29	RCL 1
30	x
31	ENTER
32	2
33	÷
34	R/S
35	GTO 07
	RUN
	g RTN

INPUT
6 R/S
1 R/S

OUTPUT	
R/S	n
R/S	s_n
R/S	Perimeter/Diameter

DESIGNING ALGEBRA EXPERIMENTS FOR THE PROGRAMMABLE CALCULATOR

Stephen L. Snover
Mathematics Department
University of Hartford
West Hartford, Connecticut 06117

As the programmable calculator (and computer) are so adept at programming routine calculations accurately and quickly, teachers should be taking advantage of their worthwhile features in the high school classroom. Not only do these calculators extend the ability to solve mathematical problems; they also bring students more quickly to the frontier of discoverable mathematics. This article describes an algebra experiment which will demonstrate the power of the programmable calculator as a function-evaluating machine.

A standard algebra II assignment is to plot the graph of $y = -(x-3)^2 + 4$, a somewhat time-consuming process. With a programmable calculator, though, the formula $-(x-3)^2 + 4$ is easily "memorized" and its values are quickly and accurately calculated, thereby allowing students to efficiently plot the graph of $y = -(x-3)^2 + 4$. Specifically, teachers might have students group in pairs to do this graphing. One student might input the values $x = 0,1,2, \ldots, 10$ into the calculator and read off the corresponding output values of $y = -5,0,3,4,3,0, \ldots, -45$, while a second student plots the indicated points $(0,-5)$, $(1,0)$ $(2,3)$ etc. on graph paper. With the increased speed and accuracy of such graphing, students will be able to advance more easily beyond the simple graphing of a function and grasp important generalizations.

The rest of this article illustrates going beyond simple graphing by showing how students can develop an understanding of the effects of the parameters a, p, and q on the graph of $y = a\cdot(x+p)^2+q$. Since the programmable calculator can memorize $a\cdot(x+p)^2+q$ just as easily as $-(x-3)^2 + 4$, students can choose their own values for a, p, and q, plot the corresponding graph, and discover for themselves the effects of their choices of parameters. The appendix at the end of this article shows in detail how to program the Texas Instruments TI 55 and Hewlett Packard HP 33E to evaluate $a\cdot(x + p)^2 + q$.

Teachers can direct students' experimentation a bit by asking them to find a, p, and q so that the graph (i) goes through the origin (0, 0), and (ii) goes through both the origin and some other point, say (10, 5).

Stephen Snover currently teaches computer science and mathematics courses to undergraduate students. He has a keen interest in mathematics education at all levels and has conducted workshops and courses for elementary and junior high school teachers on the use of computers and microcomputers. He has written several articles and books on mathematics and the teaching of mathematics.

This parameter-choosing and graphing experience can be built into an interesting game called "Catapult." Imagine a catapult located at the origin and an enemy located at the point (10, 5). The object of the game is to launch a stone from the catapult at (0, 0) and have it hit the enemy at (10, 5). Assume that the stone will follow the (trajectory) graph of $y = a \cdot (x - p)^2 + q$.

To play the game, students will need to guess a, p, and q, graph the function $y = a \cdot (x + p)^2 + q$ for $x = 0, 1, 2, \ldots, 10$ and thereby "see" the "flight" of the stone. Does the stone get launched properly? Does it hit the enemy? See Figure 1.

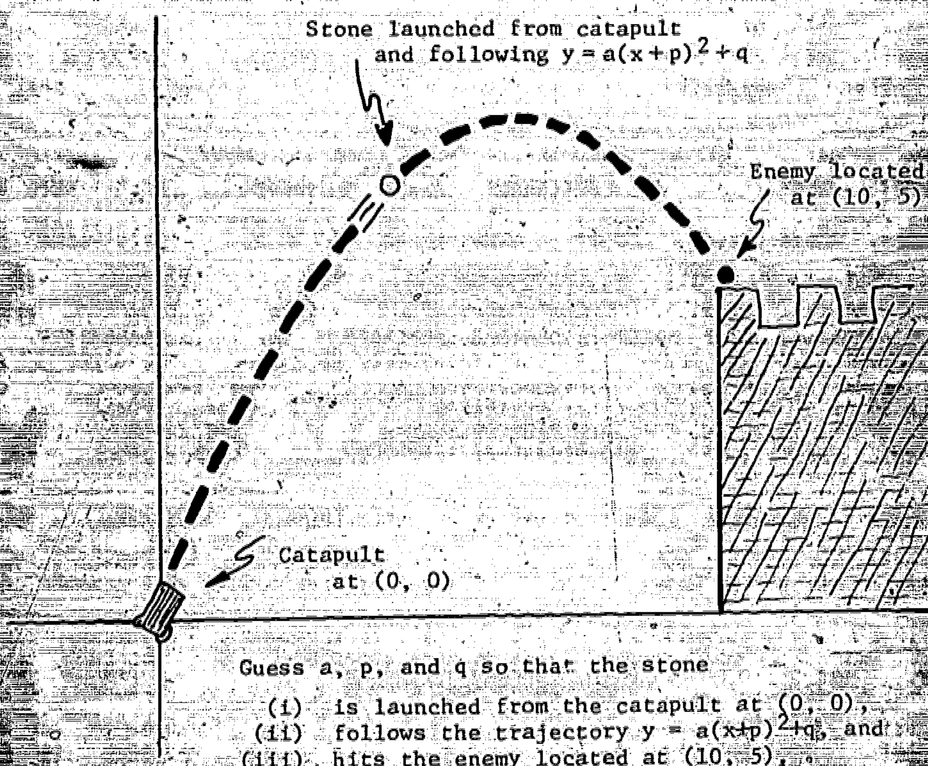


Figure 1. The Game of Catapult.

When experimenting with a, p, and q, students may try several strategies. For example, one pair of students may discover a way to make the stone reach its maximum height precisely when it hits the enemy at (10,5); i.e., have $p = 10$ and $q = 5$. They will then need to experiment with the parameter a so that the stone gets thrown from the catapult at the origin. Another pair of students may be unsuccessful at hitting the enemy and starting the stone at the origin by using $p = 5$. However, these students will learn the useful property that the trajectory has vertical symmetry around the line $x = p$, or $x = 5$ in this case, since if the stone hits (10,5) it started out at (0,5), while if it starts out at (0,0) it will hit (10,0). They may then be led to try $p = 6$ or $p = 7.5$ in order to become more successful at the game.

Whatever strategies the students come up with, sharing their experiences with others in the class can be very useful. A class discussion could conclude with statements like the following. The value of 'a' must be negative in order to make the path of the stone bend toward the ground as required by gravity. The stone will reach a maximum height at (p, q)--at least if p is between 5 and 10. If $p \leq 5$ the stone will never reach the enegy. If $p \geq 10$, the stone may hit the enemy before reaching its maximum height. For the stone to be launched from the origin, the parameter a needs to be chosen so that $a \cdot p^2 + q = 0$; i.e., so that $a = -q/p^2$.

In summary programmable calculators can be used effectively for high school algebra experiments which use them for function graphing. Students can now progress quickly beyond just the graphing of $y = -(x - 3)^2 + 4$ to the understanding of how the parameters a, p, and q affect the graph of $y = a\,(x + p)^2 + q$.

Teachers should now be designing similar experiments which will utilize programmable calculators (and computers) to advance students' knowledge and understanding of these and other algebra concepts.

APPENDIX

Programs for Evaluating $a(x+p)^2+q$ on the Texas Instruments TI 55 and the Hewlett Packard HP 33E Programmable Calculators

TI 55:

program: +, RCL, 2, =, x^2, x, RCL, 1, =, +, RCL, 3, =, 2nd R/S, 2nd RST.

sequence for inputing values of a, p, and q:
2nd RST, place value of a in display, STO, 1,
place value of p in display, STO, 2,
place value of q in display, STO, 3.

sequence for evaluating $a(x+p)^2+q$ for each value of x:
place x in display, 2nd R/S, see y in the display.

HP 33E:

program: RCL 2, +, g x^2, RCL 1, x, RCL 3, +, GTO 00.

sequence for inputing values of a, p, and q:
f PRGM, place value of a in display, STO 1,
place value of p in display, STO 2,
place value of q in display, STO 3.

sequence for evaluating $a(x+p)^2+q$ for each value of x:
place x in display, R/S, see y in the display.

A USE OF THE PROGRAMMABLE CALCULATOR IN TRIGONOMETRY: REMOVING THE AMBIGUITY FROM THE AMBIGUOUS CASE

Mary L. Johnsen
Northfield Mt. Hermon School
Northfield, Massachusetts 01360

Scientific calculators are a great boon to trigonometry students. ey remove dependence on tables and eliminate the tedium of inter- lation and logarithmic calculations. Furthermore, calculators allow w approaches to problems, approaches which were formerly impractical cause of the difficulty of the computations involved.

As an example of this, consider the Ambiguous Case: the triangle ven two sides and the angle opposite one of them. Traditionally, xtbooks solve this case by means of the Law of Sines. The biguity arises because for x such that $0 < x < 1$, there are two lues of arcsin x which are possible angles of triangles. One must sort to other considerations to determine whether both, one, or ither are legitimate solutions for the particular triangle. The biguity may be removed by using the Law of Cosines instead. This proach has not been widely used in textbooks because it leads to e solution of a very messy quadratic equation. It is rather practical unless one has access to a calculator or computer, but th a scientific calculator, the solution is quite easy. A program- ble calculator is not essential, but this is a nice problem to gram if you have one. An elegant solution requires a calculator th two memories.

Suppose that in triangle ABC we know angle A, side a, and side and we wish to find side c (if it exists). By the Law of Cosines,

$$a^2 = b^2 + c^2 - 2bc \cos A. \qquad (1)$$

is equation is quadratic in c, and can be arranged in the following rm

$$c^2 + (-2b \cos A)c + (b^2 - a^2) = 0. \qquad (2)$$

is is of the form,

$$x^2 + px + q = 0. \qquad (3)$$

the quadratic formula,

$$x = \frac{-p \pm \sqrt{p^2 - 4q}}{2} \qquad (4)$$

Mary Johnson teaches a wide range of mathematics courses from mentary algebra to advanced placement calculus at a coeducational arding school. She spent the 1979-80 academic year on sabbatical dying the use of calculators in the classroom.

An equivalent form, more useful for our purpose is

$$x = -\frac{p}{2} \pm \sqrt{\left(\frac{p}{2}\right)^2 - q} \qquad (5)$$

In equation (5) we substitute

$$x = c, \quad p = -2b \cos A, \quad q = b^2 - a^2, \qquad (6)$$

and obtain the solution

$$c = b \cos A \pm \sqrt{(b \cos A)^2 - b^2 + a^2}. \qquad (7)$$

Using

$$\sin^2 A + \cos^2 A = 1 \qquad (8)$$

and substituting in equation (7) gives the alternate form

$$c = b \cos A \pm \sqrt{a^2 - (b \sin A)^2} \qquad (9)$$

Equation (9) looks nicer, but is not any more efficient for purposes of calculation if one has a calculator with two memories. Note that only positive values of c are valid solutions. Negative, zero, and nonreal solutions must be rejected. Negative values indicate triangles containing the supplement of the given angle instead of the given angle. Nonreal values indicate that the side opposite the given angle is too short to reach the line containing the other side of the angle. Figure 1 shows some of the possible cases.

Having found the third side of the triangle, one may find the remaining angles by using the Law of Cosines. Note that if you use the Law of Sines at this stage, you might be in doubt as to whether angles are acute or obtuse.

In practical applications, one usually wants either the unknown side or the difference between the two possible values of the unknown side. In the latter case, the solution may be simplified to:

$$c - c' = 2\sqrt{(b \cos A)^2 - b^2 + a^2} \qquad (10)$$

or

$$c - c' = 2\sqrt{a^2 - (b \sin A)^2} \qquad (11)$$

The above derivations are algebraic. For those who prefer geometry to algebra, here is a geometric proof which requires only the Pythagorean Theorem and a bit of right-triangle trigonometry.

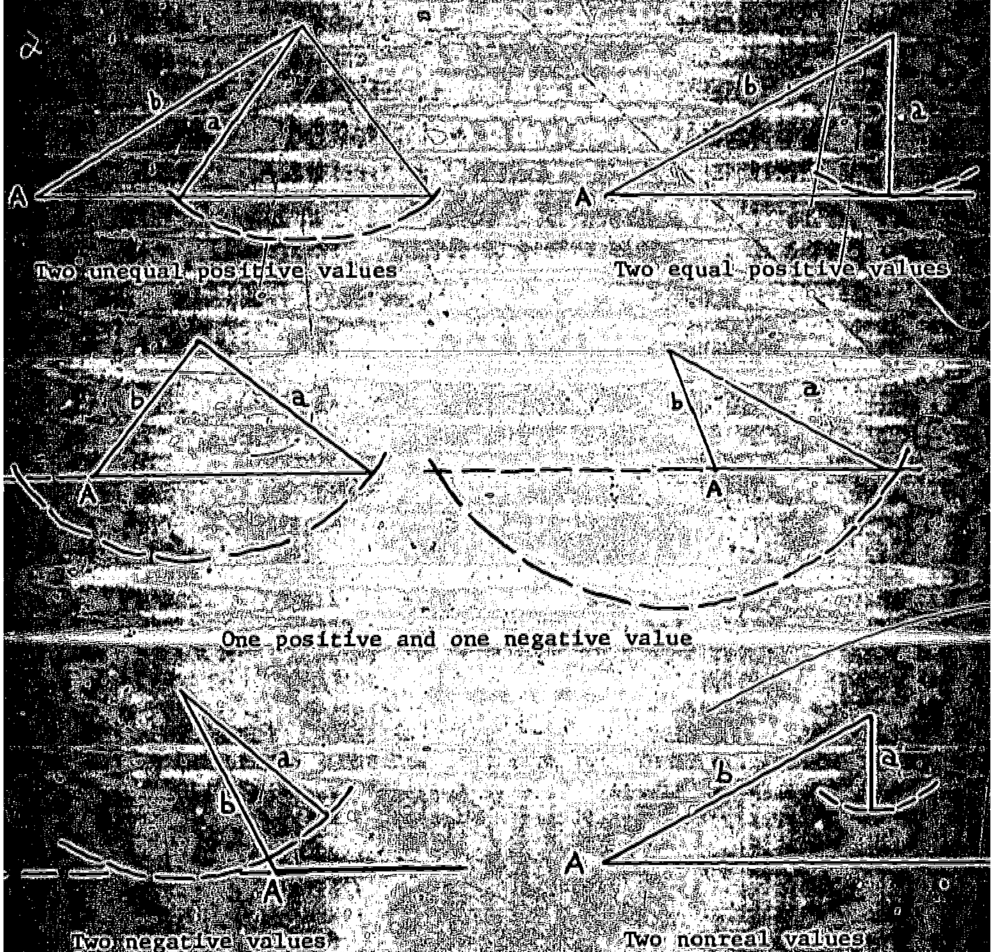

b
a
A
Two unequal positive values
b
a
A
Two equal positive values
b
a
A
b
a
A
One positive and one negative value
a
b
A
Two negative values
b
a
A
Two nonreal values

Figure 1.

In Figure 2,

$$y^2 = a^2 - h^2 \quad \text{and} \quad h^2 = b^2 - (b \cos A)^2 \tag{12}$$

So

$$y^2 = a^2 - b^2 + (b \cos A)^2 \tag{13}$$

or

$$y = \pm \sqrt{(b \cos A)^2 - b^2 + a^2} \tag{14}$$

Therefore,

$$c = b \cos A \pm \sqrt{(b \cos A)^2 - b^2 + a^2}. \tag{15}$$

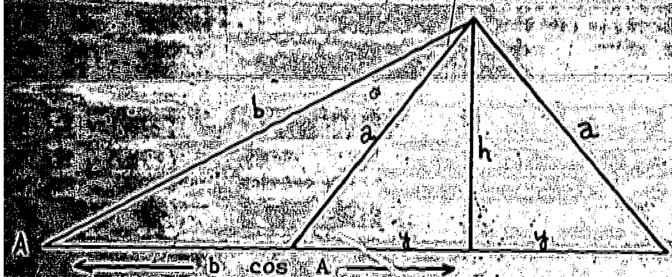


Figure 2.

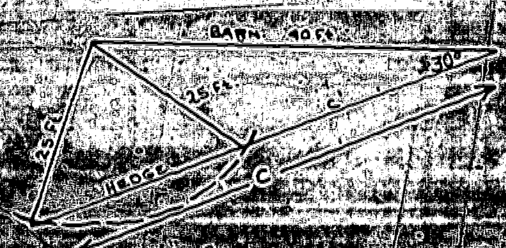
Here is a good Ambiguous Case problem: Farmer Brown's hired man tethered the goat to the corner of the barn with a 25-foot chain. From the other corner of the 40-foot long barn, a rose hedge extends for 100 feet at an angle of 30° with the barn. How much of the hedge may the goat demolish?

Solution (see Figure 3):

$$c - c' = 2\sqrt{a^2 - (b \sin A)^2}$$

$$= 2\sqrt{25^2 - (40 \sin 30°)^2}$$

$$= 30 \text{ feet}$$



For readers who have access to a TI-57 or comparable programmable calculator, here is a program designed to solve the Ambiguous Case by the method outlined above:

Store given angle in degrees in memory #0
Store opposite side in memory #1
Store other side in memory #2

00	0	12	−	24	RCL 3
01	STO 3	13	RCL 2	25	=
02	STO 4	14	x^2	26	2nd Pause
03	STO 5	15	+	27	STO 5
04	STO 6	16	RCL 1	28	RCL 3
05	RCL 0	17	x^2	29	−
06	2nd cos	18	=	30	RCL 4
07	×	19	INV 2nd x≥t	31	=
08	RCL 2	20	R/S	32	STO 6
09	=	21	$\sqrt{x}$	33	R/S
10	STO 3	22	STO 4		
11	x^2	23	+		

If solutions are nonreal, a negative number appears and the program stops, leaving zeros in memories 4, 5, and 6. Otherwise, two solutions appear and are stored in memories 5 and 6. Discard negative values. To solve another triangle, store new values in memories 0, 1, and 2, and press RST, then R/S.

USING PROGRAMMABLE CALCULATORS AND SIMULATION GAMES FOR MATHEMATICS ENRICHMENT

Michael Battista
Department of Education
Purdue University
West Lafayette, Indiana 47907

It is very common today to find various simulation games included in the hardware-software package of microcomputers and programmable calculators. Many of these games are both exciting and challenging, and can thus be very motivating to students. It is proposed here that such games can be a great way to introduce students to enrichment topics in mathematics.

To illustrate, I will briefly describe a series of activities involving a game adapted from the Texas Instruments programming manual, Making Tracks into Programming. Based on the popular television show "Battlestar Galactica," the game uses a programmable calculator to simulate a deep-space battle between a "Viper" spaceship and a mysterious Ghostship. The Viper carries four missiles with which it must dispense with the Ghostship; otherwise, the Ghostship will overcome the Viper and its pilot. A Viper pilot (the student) "shoots" his or her missiles by entering a location into the calculator. The calculator automatically displays the results. If a shot is within 4 km of the Ghostship location, it is a "hit"; otherwise, it is a miss. See the Appendix for a full description of the game.

Activities

The Ghostship game can be played by individual or small groups of students. If a TI-57 calculator is used, a person other than one of the players must enter the Ghostship location into the calculator. If a TI-58c is used, the calculator randomly generates the Ghostship location. Either way, students enjoy playing the game and generally feel challenged to develop a winning strategy.

Once students start playing the game, many opportunities for learning and doing mathematics will occur. To begin with, in order to comprehend what is happening in the game, students must understand the concept of the distance between two points and must be able to locate points using polar coordinates. In order to develop a strategy to "hit" the Ghostship with no more than four missiles,

---

Michael Battista is currently teaching mathematics methods courses for preservice elementary and secondary teachers and graduate courses in mathematics education. His research interests include the roles that spatial ability and cognitive level play in learning mathematics, problem solving, and the instructional uses of calculations and microcomputers.

students must utilize the geometry of intersecting circles. This geometry can be done at an informal level using a compass and ruler, or at a more advanced level using equations of circles or trigonometry.

After students have mastered a winning strategy, they can be asked to discover how the calculator is able to respond during the playing of the game. This involves students in analyzing both programming algorithms and mathematics, and demands some basic knowledge of trigonometry. To initiate this activity, ask students how they would calculate the distance between the Ghostship and a missile shot if they knew the polar coordinates of both points. (The calculator program uses the law of cosines as pictured in Figure 1.) Students should give their answers in the form of a set procedure or algorithm.

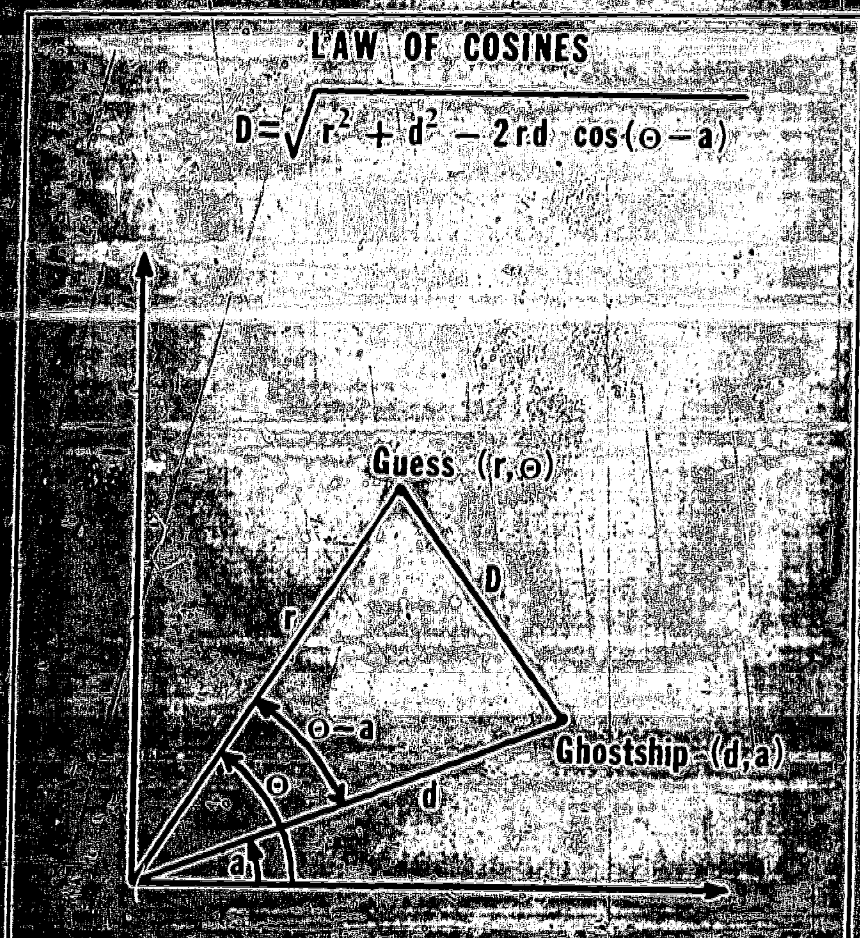


Figure 1. The law of cosines is used to calculate the distance, D, between a guess and the Ghostship.

At this point the teacher can also begin explaining the different operations that can be keyed into a calculator program. The program for the Ghostship game can be listed and explained. A flow chart may be used to give students a clearer picture of the overall flow of the program (see Figure 2). Such activities give students an opportunity to learn some of the fundamentals of programming a computing machine. Ideas universal to programming such as loops, conditional tests, and branching can be identified and discussed.

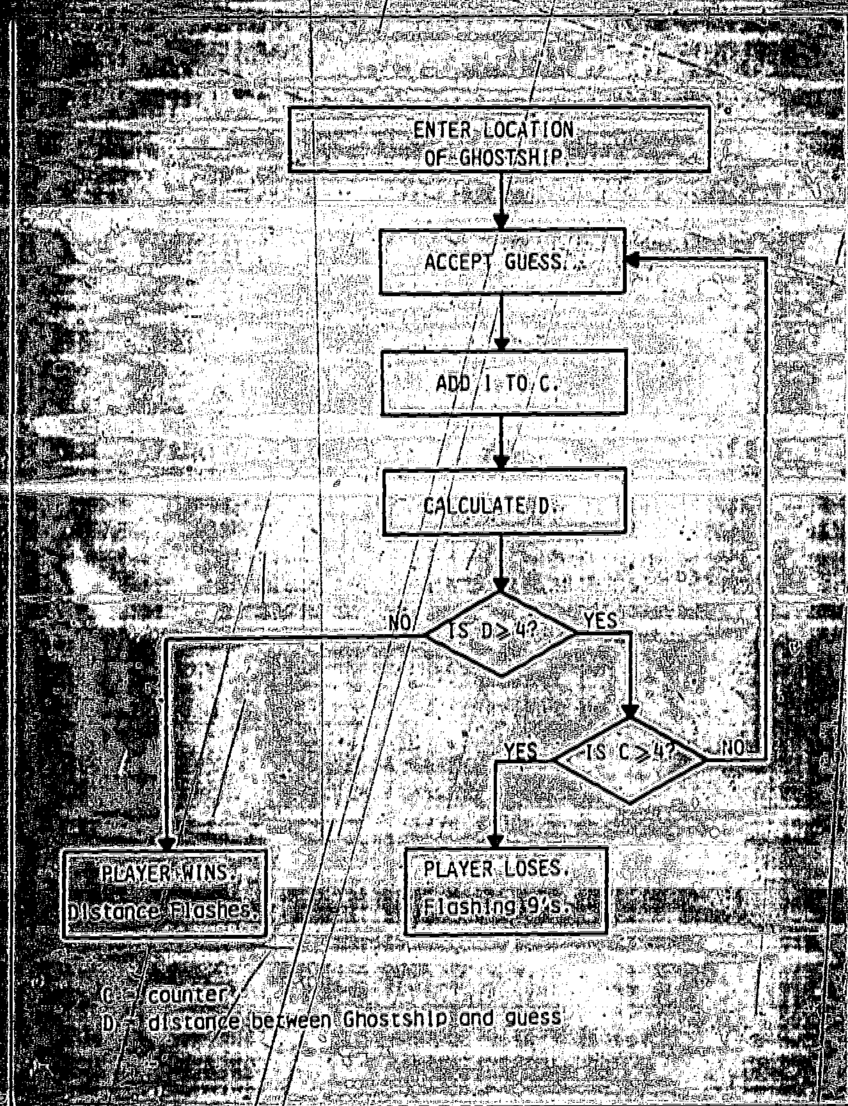


Figure 2. Flowchart for Ghostship program.

The final phase in the development of the topic is to challenge students to develop a program that will allow a second calculator automatically to find and dispense with the Ghostship. This gives students an opportunity to use mathematics to solve a real problem and to write a calculator program of their own. To meet this challenge, students must first translate their winning strategies into mathematical algorithms. This is a formidable task if the winning strategy utilized a compass and ruler. And even if students have developed a more sophisticated strategy, they will see that a great deal of clarification is required to write an algorithm that implements the strategy. As students develop their algorithms, the teacher should help them program the algorithms into a calculator.

Possibilities for extension or follow-up activities abound. The game can be played using rectangular rather than polar coordinates. This eliminates the need for trigonometry. Instead of restricting the Ghostship's location to the first quadrant, a region containing portions of all four quadrants and centered at the origin can be used. Or, the first quadrant restriction can be retained, but the outer limit on the range of the Ghostship can be extended, say from 100 km to 1000 km. The "margin of error" for hitting the Ghostship with a missile can be reduced from 4 km to 1 km. All of these rule changes, of course, require alterations in the game's calculator program. But more significantly, many of the changes require alterations in the students' strategies for winning the game, thus requiring them to delve deeper into the relevant mathematics.

Remarks

The simulation game described above is presented as a starting point for an excursion in applied mathematics. The game can be used to motivate students to study specific topics such as polar and rectangular coordinates, the geometry of intersecting circles, the law of cosines, and computer programming, or as the focal point in an enrichment unit. Numerous options exist for the teacher to choose the topics the students will investigate as they play and analyze the game. The objectives for the particular treatment described above were to have students use mathematics to solve real problems and to introduce them to programming a computing machine.

Certainly, the reader will discover other programmable calculator games that can be used to teach mathematics. In the future, these games will appear at an increasing rate. But the games themselves may be little more than amusement; they may not be educational. It is the responsibility of mathematics educators and teachers to select games that have educational value, and to design activities for using the games in ways that will promote student learning of mathematics.

Of course using a simulation game is only one way of utilizing programmable calculators to teach or enrich high school mathematics. Other uses include illustrating concepts such as functions and limits, and studying topics from numerical analysis such as iterative procedures for solving equations. There is great potential for using programmable calculators to teach students about computer programming. Few activities, however, can match the motivational effect simulation games have on students. The potential of these games actively to involve students in doing mathematics and to develop positive student attitudes towards mathematics should not go untapped.

Programmable Calculators or Microcomputers?

Which is better for use in mathematics instruction, programmable calculators or microcomputers? Obviously, the answer to the question depends on the overall objectives of the teacher. Having both available would be ideal. But, if you must choose, consider the fact that programmable calculators are almost ten times cheaper than microcomputers. You can presently buy nine or ten TI-57 calculators for the price of one TRS-80 microcomputer. For a small class, a set of ten TI-57s would offer each student significantly more hands-on experience than a single microcomputer. So, in these times of shrinking budgets, it seems that, as an instructional aid for teaching mathematics, programmable calculators offer a viable alternative to microcomputers.

Note on Models of Calculators

The program listed in the Appendix is for a TI-58 calculator. Although playing the Ghostship game on the more powerful TI-58 is more convenient than on a TI-57, I would recommend the TI-57 for high school use. I say this, first, because the TI-57 is a simpler machine to program and understand, and second, because the programming manual for the TI-57, Making Tracks into Programming, is much more readable than the TI-58 manual.

A program for playing "Ghostship" on a TI-57 is given in the TI-57 programming manual. It requires only slight modification in order to be used with the game rules given here.

APPENDIX

Description of the Game
(To be given to students)

Imagine yourself a character in a science fiction tale such as Star Wars or Battlestar Galactica. The setting is deep space, and you are a "Viper" pilot. A Viper is your planet's most advanced fighter spaceship. In a certain sector of space a strange phenomenon has been occurring for years. The radar system of a Viper will detect another ship within range of the Viper's attack missiles. But, strangely enough, the Viper's sighting screen will not detect the ship, so its whereabouts will be unknown. Thus the name "Ghostship" has been given to this eerie type of hostile ship.

And hostile they are. They have been attacking your planet's ships for years. But they do it in a rather unique way. A Ghostship seems to lock on to a Viper, staying the same distance away and in the same relative position. The Ghostship shoots some sort of slow working rays at the Viper which eventually render the Viper inoperative. The Ghostship then simply lets the Viper float in space--forever.

The only defense a Viper has against these so-called Ghostships is its battery of four attack missiles. The missiles are shot at a point at which the Ghostship is thought to be located. If the shot is within 4 km of the ghostship, the ghostship is eliminated. But if all the shots miss, the demise of the Viper is imminent.

Now down to business. Your calculator is exactly like the attack-missile control in a Viper. The ghostship is somewhere at a range of 0 to 100 km, and between the angles of 0 and 90° on your target locater chart. To shoot an attack missile, choose a distance and an angle. (Note how to locate this point on your locater.) Enter the distance on your calculator and press R/S. Then enter the angle and press R/S. If you hit the Ghostship, the number of kilometers that you missed by will flash on the calculator. (Then you're safe.) If you miss, the distance that you missed by is displayed. (Your ship's radar is capable of giving this information to you.) To fire your next missile, repeat the above procedure. Remember, you have only four shots. If by the fourth shot you have failed to hit the Ghostship, the calculator display will flash "99999999 99" and you have lost the battle.

Program Description

The Ghostship's location in polar coordinates (d, a) is randomly picked by the calculator. The limits on the parameters are: $0 \leq d \leq 100$, $0 \leq a \leq 90$. A player has four chances to guess the location of the Ghostship within four kilometers. If the player "hits" the Ghostship, the actual distance of the miss is flashed on the display. If the player misses but has shots remaining, the distance of miss is displayed. If the player fails to hit the Ghostship after four shots, 9s will flash on the display.

User Instructions

(To be followed after the program has been keyed in.)

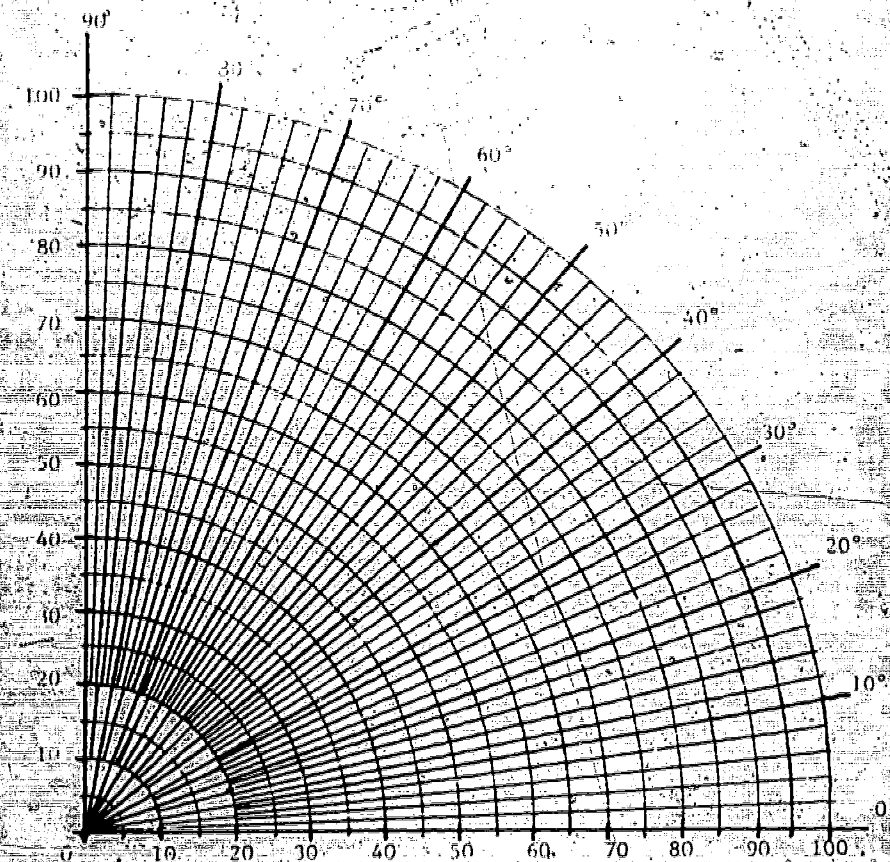
STEP	PROCEDURE	ENTER	PRESS	DISPLAY
1	Clear memories.		2nd, CM's	
2	Initialize random generator.		2nd, Pgm, 15 2nd, E'	0.
	Enter "seed."	Any number	E	The number.
3	Reset program to location 000.		RST	Same.
4	Run program.		R/S	0
5	Now enter your guesses for the Ghostship location —distance first, angle second.	Distance, d Angle, a	R/S R/S	 Result.
6	To play the game again, press CLR and return to step one.			

Program for TI-58 Calculator

LOCATION	CODE	KEY	COMMENTS
000	36	2nd, Pgm	
001	15	15	
002	71	SBR	
003	88	2nd, D.MS	
004	65	x	Enters location of Ghostship (d, a) using random number generator from library.
005	01	1	
006	00	0	
007	00	0	
008	95	=	
009	42	STO	
010	00	00	
011	36	2nd, Pgm	
012	15	15	
013	71	SBR	
014	88	2nd, D.MS	
015	65	x	
016	09	9	
017	00	0	
018	95	=	
019	42	STO	
020	01	01	
021	00	0	
022	91	R/S	
023	42	STO	
024	02	02	Accepts guess--distance first, then angle.
025	91	R/S	
026	42	STO	
027	03	03	
028	04	4	
029	32	x ⇄ t	Puts "4" into the t register. Register 4 is used as a "counter."
030	01	1	
031	44	SUM	
032	04	04	
033	43	RCL	
034	03	03	
035	75	-	
036	43	RCL	
037	01	01	
038	95	=	
039	39	2nd, COS	
040	65	x	Calculates distance, D, between guess and Ghostship using the law of cosines. Stores this distance in register 5.
041	43	RCL	
042	00	00	
043	65	x	
044	43	RCL	
045	02	02	

LOCATION	CODE	KEY	COMMENTS
046	65	x	
047	02	2	
048	94	+/-	
049	85	+	
050	43	RCL	
051	00	00	
052	33	x^2	
053	85	+	
054	43	RCL	
055	02	02	
056	33	x^2	
057	95	=	
058	34	$\sqrt{x}$	
059	42	STO	
060	05	05	
061	77	2nd, $x \geq t$	
062	11	A	Test: If D≥4, go to label A. If not, go to "+ +." Since the double plus is illegal, calculator display flashes.
063	85	+	
064	85	+	
065	91	R/S	
066	76	2nd, Lbl	
067	11	A	
068	43	RCL	
069	04	04	Recall number of times through program. If it is greater than or equal to 4, go to end at label B. Otherwise, display D, accept new guess, and start over at location 023.
070	77	2nd, $x \geq t$	
071	12	B	
072	43	RCL	
073	05	05	
074	91	R/S	
075	61	GTO	
076	00	0	
077	23	23	
078	76	2nd, Lbl	
079	12	B	Label B, end. Dividing by zero is illegal and results in flashing 9s.
080	00	0	
081	35	1/x	
082	91	R/S	

TARGET LOCATER



USING THE PROGRAMMABLE CALCULATOR FOR SIMULATION--A PROBABILITY EXPERIMENT

Dave Haggerty
Monroe High School
Monroe, Oregon 97456

The programmable calculator can be used in many different ways in the classroom. One way is for simulation. The calculator can be programmed to run simulated experiments that would be cumbersome or time-consuming. For example, it can be programmed to roll 1, 2, 3, or more dice, total them, count the number of times a particular outcome is rolled, and stop after a specified number of rolls.

We used the calculator to simulate the following probability experiment. We considered a setup of three stacks of blocks; four blocks in the first stack, six in the second, and eight in the third. We decided to have one of the stacks chosen at random and from that stack one block removed. This process was to be repeated until one of the stacks was depleted, thus ending the experiment.

We programmed the HP 33E calculator with the program presented at the end of this article and found that each experiment took about 15 seconds. By pressing RCL 1, RCL 2 and RCL 3, we could see and record how many blocks were taken from each stack. Also by pressing RCL 4 we obtained the total number of blocks taken from all the stacks. Once the data were recorded, the experiment could be run again by pressing R/S.

When we did this in the classroom, each of the 10 students ran the experiment 20 times, recorded their data, and answered the following questions. (These are but some of the many related questions that could be asked.)

1. What was the mean (average) number of blocks taken on each run?

2. What was the mean number of blocks taken from stack #1? ...from stack #2? ...stack #3?

3. How many times was stack #1 depleted before the other stacks? ...stack #2? ...stack #3?

Dave Haggerty is the mathematics faculty at Monroe High, a small rural school in a farming and lumber area of Oregon. He teaches courses from general math through precalculus and uses programmable calculators in algebra and computer science courses. He has conducted a workshop on programmables for secondary teachers at a statewide teachers' conference.

4. Using your data, what is your experimental probability that a total of 10 blocks will be taken before one of the stacks is depleted?

5. What is your experimental probability that stack #1 will be depleted?

Once the students completed this exercise all of the data were compiled and the class answered similar questions for the entire 200 runs. The students then compared their results against the totals. Using the programmable calculator in this way, the class was able to compile the composite results of 200 experiments in one class period, a task that would not have been possible to do without a programmable calculator or similar machine. Furthermore, the 200 experiments were sufficient for us to obtain experimental probabilities that could be trusted for prediction purposes.

With the program we wrote for the HP-33E, a whole range of similar probability experiments could be simulated. By inserting different numbers in steps 26, 31, and 36 of the accompanying program (see Appendix), the number of blocks in each of the three stacks can be changed. You can experiment if you like and see what effect these changes have on the outcome.

The kinds of simulation experiments programmable calculators can be programmed to perform are endless. Moreover, because of their efficiency, simulations previously impossible to do in a classroom setting because of time considerations become very do-able.

APPENDIX

PRGM Mode

KEYSTROKE	DISPLAY	KEYSTROKE	DISPLAY
1	01- 1	1	28- 1
STO + 4	02-23 51 4	STO + 2	29-23 51 2
RCL 0	03- 24 0	RCL 2	30- 24 2
RCL 7	04- 24 7	6	31- 6
x	05- 61	GTO 37	32- 13 37
g frac	06- 15 33	1	33- 1
STO 0	07- 23 0	STO + 1	34-23 51 1
1	08- 1	RCL 1	35- 24 1
0	09- 0	4	36- 4
x	10- 61	f x≠y	37- 14 61
g int	11- 15 32	GTO 01	38- 13 01
g x=0	12- 15 71	RCL 4	39- 24 4
GTO 03	13- 13 03	R/S	40- 74
.	14- 73	0	41- 0
3	15- 3	STO 1	42- 23 1
x	16- 61	STO 2	43- 23 2
g int	17- 15 32	STO 3	44- 23 3
g x=0	18- 15 71	STO 4	45- 23 4
GTO 33	19- 13 33	GTO 01	46- 13 01
1	20- 1		
-	21- 41		
g x=0	22- 15 71		
GTO 28	23- 13 28		
STO + 3	24-23 51 3		
RCL 3	25- 24 3		
8	26- 8		
GTO 37	27- 13 37		

RUN Mode

1. Enter a 6-digit decimal fraction, last digit being odd but not 5, and store it in the 0 register. STO 0
2. Enter 147 and store it in the 7 register. 147 STO 7
3. Return the program to the beginning. g rtn
4. Data are now ready to be generated. R/S
5. After each run, push the R/S button and new data will be furnished.

INDEPENDENT STUDY WITH A PROGRAMMABLE CALCULATOR

Lee Mohler
Department of Mathematics
University of Alabama in Birmingham
Birmingham, Alabama 35294

Electronic computing devices have not yet made significant inroads into the high school mathematics curriculum, although there is beginning to be widespread agreement that they should. It is clear that computers are revolutionizing scientific practice (along with just about everything else); thus, if the schools are to realistically prepare students for the technological world of the '80s, they must come to grips with electronic computation. This paper presents a sample format for using programmable calculators in independent study.

The following material is an excerpt from Mathematical Recreations for the Programmable Calculator (Mohler and Hoffman, 1981), a collection of programming problems designed to teach the standard techniques of programming. The problems presented here involve the use of nested loops, a technique introduced to students in a previous chapter of the book. The student is to read the statements of the problems and attempt to solve them before looking at the solutions. The solutions themselves are quite complete, but do not include program listings. The idea is to encourage the student to do as much as possible before looking at the solution. The student who works through this material will learn all that is needed about nested loops. Exercises are, after all, the heart of the learning process in mathematics. The goal of the book and the problems given here is to present programming exercises within a sufficiently interesting setting that students will be stimulated to want to solve them on their own.

It is worth noting that this particular excerpt contains a good deal of mathematical history. Aside from its intrinsic interest, this material promotes the idea that mathematics is a creation of human beings and is even now developing and changing. Ambitious students may one day hope to make their own mark on that history.

Subscripted m's appearing in the flowcharts for the solutions of the problems represent memories. The arrow notation means roughly "replace by." For example, $m_1 \leftarrow m_0 + 1$ means add 1 to the contents of m_0 and store the result in m_1 (the contents of m_0 remains unchanged).

Lee Mohler is a research topologist with an active interest in the history of mathematics. He is currently developing materials for programmable calculators to be used in the calculus curriculum.

Pythagorean Triples

Recall the famous Pythagorean theorem: If a, b, and c are the legs and hypotenuse of a right triangle (see Figure 1), then $a^2 + b^2 = c^2$. If a, b, and c are (positive) whole numbers satisfying this equation, then a, b, c is called a Pythagorean triple. The simplest example is the triple 3, 4, 5. You are probably already familiar with the fact that $3^2 + 4^2 = 5^2$. Pythagorean triples have fascinated mathematicians for millenia (see the Notes). There are lots of them --infinitely many to be precise--but they are rather thinly scattered. The next interesting one after 3, 4, 5 is 5, 12, 13 ($5^2 + 12^2 = 25 + 144 = 169 = 13^2$).[1]

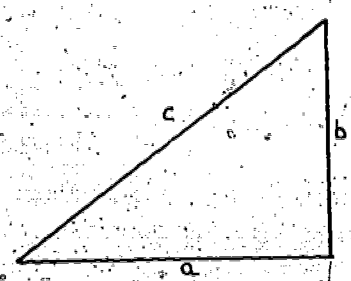


Figure 1

Problem 1: Write a program which searches out and finds all Pythagorean triples. The program should generate triples in some systematic way, screening for and outputting only the Pythagorean ones.

If you have solved Problem 1 and run your program, you will have noticed how slow it is, and that the larger the numbers in the triples get, the longer it takes the calculator to find them. You can imagine, then, how hard it would be for humans to find Pythagorean triples by the method of searching! So a long time ago certain clever individuals began looking for formulas which would generate Pythagorean triples automatically. Here is a scheme attributed to Pythagoras himself:

Pythagoras noticed that the sum of all the odd numbers up to some point always added up to a perfect square: $1 + 3 = 4 = 2^2$, $1 + 3 + 5 = 9 = 3^2$, $1 + 3 + 5 + 7 = 16 = 4^2$, etc. Now suppose the last term in such a sum is itself a perfect square, as in the sum $1 + 3 + 5 + 7 + 9$. Then the whole sum is a perfect square (25 in this case) and can be broken into two other perfect squares (namely

[1] An uninteresting one is 6, 8, 10, which is just the 3, 4, 5 triple "blown up" by a factor of two.

the sum up to but not including the last term, and the last term by itself), thereby creating a Pythagorean triple:

$$25 = 1 + 2 + 3 + 5 + 7 + 9 = (1 + 3 + 5 + 7) + (9) = 16 + 9 = 4^2 + 3^2.$$

The next sum for which this holds true works is $1 + 3 + 5 + \cdots + 21 + 23 + 25$:

$$13^2 = 169 = 1 + 3 + 5 + \cdots + 21 + 23 + 25 = (1 + 3 + 5 + \cdots + 21 + 23) + (25) = 144 + 25 = 12^2 + 5^2,$$

yielding the Pythagorean triple 5, 12, 13.

It becomes clear that <u>any</u> odd perfect square, m^2, sits at the end of a sum of consecutive odd numbers, $1 + 3 + 5 + \cdots + (m^2 - 2) + m^2$. We will spare you the details which show that $1 + 3 + 5 + \cdots + (m^2 - 2)$ adds up to $\left(\frac{1}{2}(m^2 - 1)\right)^2$ and that the whole sum $1 + 3 + 5 + \cdots + (m^2 - 2) + m^2$ adds up to $\left(\frac{1}{2}(m^2 + 1)\right)^2$, producing the Pythagorean triple m, $\frac{1}{2}(m^2 - 1)$, $\frac{1}{2}(m^2 + 1)$. Thus we have Pythagoras' formula: If m is any odd number, then m, $\frac{1}{2}(m^2 - 1)$, $\frac{1}{2}(m^2 + 1)$ is a Pythagorean triple. Here is a problem we leave you to work on your own (no solution is provided).

<u>Problem 2</u>: Write a program which generates Pythagorean triples, using Pythagoras' formula.

Pythagoras' scheme does not generate all Pythagorean triples. If you have solved Problem 2, you have probably noticed that the program produces only triples a, b, c where $c = b + 1$ (i.e., $\frac{1}{2}(m^2 + 1) = \frac{1}{2}(m^2 - 1) + 1$). Thus it will not generate the triple 8, 15, 17, which you can readily verify to be Pythagorean. What is needed is a formula taking two numbers as input instead of one. This will provide enough flexibility to produce all Pythagorean triples.

Here it is: Let z and w be any two positive whole numbers with $w < z$. Then $a = z^2 - w^2$, $b = 2zw$, and $c = z^2 + w^2$ is a Pythagorean triple (proof: $a^2 + b^2 = (z^2 - w^2)^2 + (2zw)^2 = z^4 - 2z^2w^2 + w^4 + 4z^2w^2 = z^4 + 2z^2w^2 + w^4 = (z^2 + w^2)^2 = c^2$). For example, let $z = 7$ and $w = 4$. Then $a = 49 - 16 = 33$, $b = 2 \cdot 7 \cdot 4 = 56$, and $c = 49 + 16 = 65$. You can check on your own that 33, 56, 65 is a Pythagorean triple. Moreover, this scheme generates all but a few uninteresting triples. (It does not, for instance, generate the triple 9, 12, 15, which is just the triple 3, 4, 5 "blown up" by a factor of 3.)[2] It does get all of the "primitive" ones (i.e., ones which are not multiples of others).

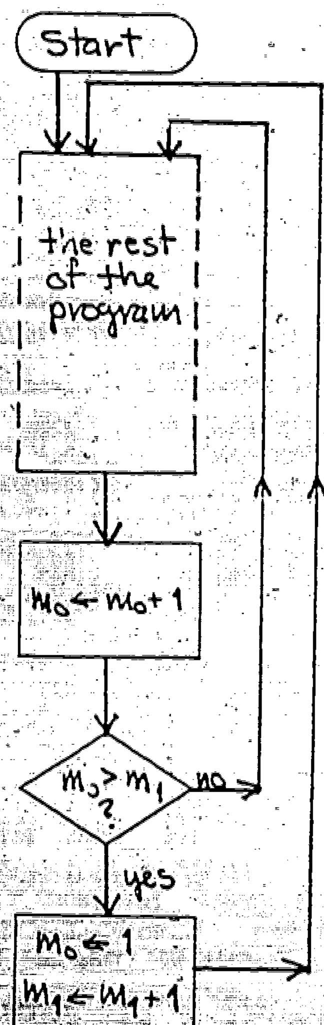
[2] Unfortunately, it does not avoid all of the uninteresting triples. Note this exception, among others: if $z = 3$ and $w = 1$, we get the uninteresting triple 6, 8, 10. See the Notes for more on the subject.

Problem 3: Write a program for generating all Pythagorean triples, using the preceding formulas.

Solutions

Problem 1: Notice that if $a^2 + b^2 = c^2$, then $c = \sqrt{a^2 + b^2}$. Thus the Pythagorean triple a, b, c can be rewritten a, b, $\sqrt{a^2 + b^2}$. The idea of the solution is to generate all triples a, b, $\sqrt{a^2 + b^2}$, checking each time to see if $\sqrt{a^2 + b^2}$ is a whole number. If it is, then a, b, $\sqrt{a^2 + b^2}$ is a Pythagorean triple, and the program stops to output it. We may as well only generate those triples for which $a \leq b$, since the others are redundant (the triple 4, 3, 5 is no different from the triple 3, 4, 5).

So first we need a part of the program which generates all possible pairs of whole numbers a, b with $a \leq b$. This we accomplish with nested loops. Here is the flowchart:



Memories

0	a
1	b

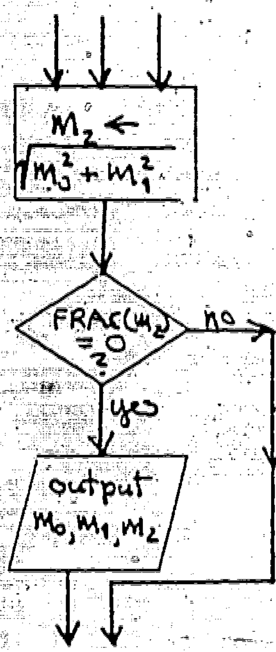
Initial State of Memories

0	1
1	1

or

0	wherever you
1	want to start

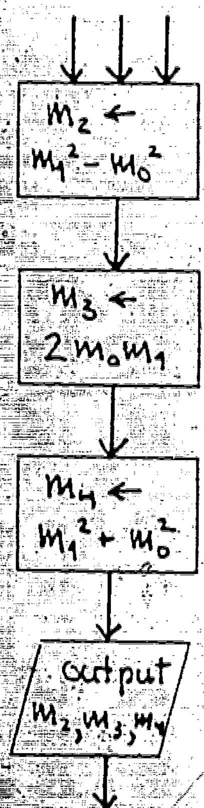
Now all the rest of the program has to do is compute $\sqrt{a^2 + b^2}$, store it in a memory (in case it turns out to be a whole number), and check to see if it is a whole number (fractional part = 0?). If it is, output the triple a, b, c. We will leave it to you to arrange the output. If $\sqrt{a^2 + b^2}$ is not a whole number, it is time to generate a new pair a, b, which takes you into the part of the program we have already described. Here is the flowchart for the rest of the program:



Memories

0	a
1	b
2	c

Problem 3: In this program we want to generate all pairs z, w with w < z; and to generate from each pair z, w the triple a, b, c, using the formulas given before the statement of the problem. The part of the program generating the pairs z, w will be just like the scheme for generating the pairs a, b in the previous solution, except that the test $m_0 > m_1$? gets replaced by the test $m_0 = m_1$? The flowchart for the rest of the solution looks like this:



Memories

0	w
1	z
2	a
3	b
4	c

Initial State of Memories

0	1
1	2
2	
3	
4	

(Register arithmetic can be used to advantage in this program.)

As noted earlier, there is much history associated with this material. Students might find the following of interest. Pythagoras got his name attached to the Pythagorean theorem and Pythagorean triples more or less by historical accident. Until the 20th century, it had been thought that there was very little mathematics worth mentioning before the time of Pythagoras. However, we now know that the Egyptians and Babylonians were producing respectable mathematics as far back as 2000 B.C. (1400 years before Pythagoras!). Indeed, the more we learn about their work (especially that of the Babylonians), the more impressive it becomes. Clay tablets dating from about 1800 B.C. show that Babylonians of that period knew how to generate Pythagorean triples.

As we noted earlier, the scheme for generating Pythagorean triples given in Problem 3 produces all the "primitive" triples, the ones from which all others can be obtained by multiplication. For example, 5, 12, 13 is primitive. From it we can generate the triples 10, 24, 26 (multiply by 2); 15, 36, 39; 20, 48, 52; etc. But 5, 12, 13 is not itself a multiple of any other Pythagorean triple. Clearly if we are interested in generating all Pythagorean triples, it suffices to generate only the primitive ones. The scheme given in Problem 3 does not do this. In addition to the primitive triples, it generates some (though not all) nonprimitive ones. The nonprimitive triples can be avoided by imposing two restrictions on the numbers z and w: (i) they should have no common divisor other than 1 (i.e., they should be relatively prime); and (ii) they should not both be odd (they cannot both be even either, since then 2 would be a common divisor). It is tricky to write a program embodying these restrictions; we leave it to the fanatics among our readers.

Pythagorean triples can be generalized in various ways. There are, for example, "Pythagorean quadruples," three perfect squares adding up to a perfect square, such as 6^2, 10^2, 15^2, and 19^2. One can also find quadruples of perfect cubes ($3^3 + 4^3 + 5^3 = 6^3$), quintuples of perfect fourth powers ($30^4 + 120^4 + 272^4 + 315^4 = 353^4$), sextuples of perfect fifth powers, etc. However, there are no pairs of cubes adding up to a perfect cube; i.e., there are no triples of (positive) whole numbers a, b, and c such that $a^3 + b^3 = c^3$. An incomplete proof of this fact was given by Euler in the 18th century. In the 19th century Gauss, perhaps the greatest mathematician of all time, gave the first entirely correct proof.

Students who work through this material will now know how to set up a nested loop. Hopefully they will also have developed some appreciation for the beauty of mathematics and the power of modern electronic technology.

Reference

Mohler, Lee and Hoffman, Dean. Mathematical Recreations for the Programmable Calculator. New York: Hayden, 1981.

USING THE PROGRAMMABLE CALCULATOR TO ENHANCE SCHOOL MATHEMATICS LEARNING AND INSTRUCTION IN THE 1980s: ONE SUGGESTION

J. F. Weaver
Department of Curriculum and Instruction
The University of Wisconsin-Madison
Madison, Wisconsin 53706

I wish to suggest a particular use for programmable calculators at amalgamates (a) a widely accepted interpretation of the nature mathematics as a discipline, (b) a significant idea or concept at permeates much of mathematics, and (c) an important aspect of thematical learning and instruction.

A. Mathematics As a Discipline

Mathematics, including *school* mathematics, can be (and has been) aracterized in a variety of ways. Within the past several decades, wever, there seems to be more than a modicum of agreement on the ture of mathematics, at least in relatively broad terms. For stance:

More than 20 years ago in one of his treatises Sawyer (1955) ated:

> For purposes of this book we may say, "Mathematics is the classification and study of all possible patterns." Pattern . . . is to be understood in a very wide sense, to cover almost *any kind of regularity that can be recognized by the mind*. (p. 12)

Approximately 15 years later, Lovell (1971) contended that

> ...we can do no better than to follow the Bourbaki group of mathematicians' interpretation of mathematics as the study of structures or the study of *systematic patterns of relationship*. (p. 21, italics added)

More recently, Ginsburg (1977) contended that

> Children must learn not only to calculate, but to *perceive*, to *see*. They must learn that mathematics is

---

J. Fred Weaver is a mathematics educator who has made numerous ntributions to the profession through his research, workshops, lks, and publications. He currently works with preservice and -service teachers and with advanced graduate students in mathe- tics education, K-12.

> concerned with important regularities, and that numbers behave in orderly ways that they can predict. (p. 160)

B. A Significant Mathematical Idea or Concept

Many persons (e.g., Stone, 1965; Buck, 1975) have discussed the significance of the idea or concept of *function*[1] in connection with the discipline of mathematics. Its role within *school* mathematics programs has been suggested and illustrated in a variety of ways (e.g., Cambridge Conference on School Mathematics, 1963; Wirtz, Botel and Nunley, 1963; Davis, 1964, 1967; Page, 1964; Karlin, 1965; Anderson, 1967; Esty, 1967), which include pictured "function machines." Its inclusion (to some degree) within school mathematics text series as low as the *elementary* level can be traced back almost 20 years (e.g., Wirtz and Botel, 1961). And the development of the function idea or concept among school students has been the subject of more than one research investigation (e.g., Orton, 1970; Nunley, 1973; Thomas, 1975).

C. An Important Aspect of Mathematical Learning and Instruction

Gagne's (1977) identification of different types of learning applies not only to learning in general, but to mathematical learning in particular. In Avital and Shettleworth's terms, an important aspect of mathematical learning and instruction involves the process of *open search*. This is akin to Wittrock's (1974) hypothesis:

> Succinctly, but abstractly stated, . . . that human learning with understanding is a generative process, involving the construction of (a) organizational structures for storing and retrieving information, and (b) processes for relating new information to the stored information.
>
> Stated more directly, all learning that involves understanding is discovery learning. (p. 182)

Wittrock's hypothesis is not unlike Scandura's (1971) "generative procedures," and his emphasis upon "discovery" has many familiar counterparts in the educational psychology and mathematics education literature, including Wirtz's (1963) concern for "nonverbal instruction."

Use of the programmable calculator as a "real, live *function machine*" can provide an excellent means of (a) directing attention to the underlying nature of mathematics (b) with a focus upon a significant mathematical idea or concept (c) through a process that exemplifies an important kind or type of mathematical learning and instruction—ranging from relatively simple to more sophisticated content contexts.

Some Illustrations

I have appended several suggested content contexts that, I feel, an be used to advantage in implementing the preceding point of view. y own programs were written for an HP 41C calculator with a eripheral printer (having alphanumeric capabilities). [Editor's ote: In an end note to this article, Professor Weaver invites eaders to write to him for actual programs for a variety of calcu-ators.] Record sheets have been used along with the calculator/rinter-generated hardcopy or printout associated with the first hree of the following four problem contexts (all presented in the ppendix):

1. SIXRULE (Supplementary use of graphical representation not included.)

2. FOURULE

3. TRIRULE

4. PYTRU (somewhat different from and more "sophisticated" than the preceding contexts)

Except for an example or two to get started in a particular tuation with input suggested by the instructor, students should ve opportunity to select their own input values, and to modify ccessive inputs in accord with the output generated. One vantage of a calculator such as the HP 41C with its peripheral inter is the printout of error messages in the case of invalid put.

What strategies do students devise and use in their search for rule? In connection with SIXRULE situations, for instance, do udents of their own volition move in the direction of a "finite fferences" approach (Seymour and Shedd, 1973)? What kind and gree of guidance can/should instructors give students in their arch for a rule?

In what verbal or other "more mathematical" form do students ecify rules? How precisely are they specified?

When a rule seemingly has been "discovered," how well are udents able to cope with "inverse examples": what input must used to generate a particular output?

The preceding questions signify just a few of the things to be pt in mind when working with students on "What's My Rule?" and milar problem situations, which seem to have considerable positive tivational appeal.

Finally, I don't believe that I am a sadist--but there are times en I prefer that instructors rather than students select input lues in a deliberate attempt to mislead students into generating n inadequate or insufficient rule, as illustrated in Figure 1 in onnection with a FOURULE situation. There is much to be learned t times from being misled!

In connection with the appended material, the *calculations* nherent in each of the four contexts (SIXRULE, FOURULE, TRIRULE, nd PYTR) are relatively simple. Programs are longer than might e expected, however, in order to accommodate decisions regarding he validity of input values and the printing of error messages n instances of invalid input or output.

Although I have made program modifications for use with calcu-ators *without* accompanying printers, I much prefer calculator/ rinter combinations--particularly those with *alpha* as well as umeric capabilities--just as I find it advantageous to use alculators with nonvolatile program memories or with magnetic ard-reading capabilities.

End Notes

. I will be glad to send a copy of the FOURULE or TRIRULE program for the HP 41C (with 82143A printer) calculator to any person requesting such. Write to J. F. Weaver, Department of Curriculum and Instruction, University of Wisconsin-Madison, 225 North Mills Street, Madison, WI 53706.

. The same applies to SIXRULE which I now have in a newer MULTI-RULE version for HP 41C (and printer), HP-97, HP-67, HP-65, HP-29C, HP-25C, HP-19C, Texas Instruments SR-52 (with printer) and TI-59 (with printer). (A version of MULTIRULE also has been prepared in BASIC for my PET microcomputer.)

. In the June/July/August 1980 issue of HP KEY NOTES (Vol. 4, No. 2) I illustrated use of the MODulo function in connection with the Euclidean algorithm. The same function could have been used in my PYTR program to reduce program length by a few lines. (This would be true also for my FOURULE and TRIRULE programs.)

. A version of PYTR has been prepared in BASIC for my PET micro-computer.

. Some additional PYTR notes are available as separate sheets, if desired. These pertain to calculation procedures, use of data storage, registers and flags, and similar considerations.

RULE A	
8, 6 → 2	Based upon input/output data fo
15, 12 → 3	trials 1 thru 9, it would be plausibl
20, 16 → 4	to conclude that Rule A is, in effect
21, 14 → 7	$x, y \longrightarrow x - y$
26, 25 → 1	
27, 18 → 9	
32, 24 → 8	
36, 30 → 6	
45, 40 → 5	
10, 15 → 5	Trials 10-18, however, would pro
12, 18 → 6	some modification in the previously c
16, 20 → 4	jectured rule--now phrased possibly i
21, 24 → 3	terms of "differences" and maybe
28, 35 → 7	absolute values.
31, 32 → 1	
32, 34 → 2	
36, 45 → 9	
40, 48 → 8	
16, 10 → 2	And trials 19-21, for instance,
20, 35 → 5	would cause one to reject the origina
24, 24 → 24	conjecture or its subsequent modifica-
100, 49 → 1	tion.

Figure 1. FOURULE program.

References

Allendoerfer, C. B. and Oakley, C. O. *Principles of Mathematics* (2nd ed.). New York: McGraw-Hill, 1963.

Anderson, R. C. Let's Consider the Function! *Arithmetic Teacher* 14: 280-284, 1967.

Avital, S. M. and Shettleworth, S. J. *Objectives for Mathematics Learning: Some Ideas for the Teacher* (Bulletin No. 3). Toronto: Ontario Institute for Studies in Education, 1968.

Buck, R. C. Functions. In E. G. Begle (Ed.), *Mathematics Education* (69th Yearbook, Part I, National Society for the Study of Education). Chicago: University of Chicago Press (for the Society), 1970.

Cambridge Conference on School Mathematics. *Goals for School Mathematics.* Boston: Houghton Mifflin (for Educational Services, Inc.), 1963.

Davis, R. B. *Discovery in Mathematics.* Palo Alto, California: Addison-Wesley, 1964.

Davis, R. B. *Explorations in Mathematics.* Palo Alto, California: Addison-Wesley, 1967.

Esty, E. Functions. *Arithmetic Teacher*, 14: 657-664, 1967.

Gagne, R. B. *The Conditions of Learning* (3rd ed.). New York: Holt, Rinehart & Winston, 1977.

Ginsburg, H. *Children's Arithmetic: The Learning Process.* New York: Van Nostrand, 1977.

Karlin, M. Machines. *Arithmetic Teacher* 12: 327-334, 421, 1965.

Lovell, K. Some Aspects of the Growth of the Concept of a Function. In M. F. Rosskopf, L. P. Steffe, and S. Taback (Eds.), *Piagetian Cognitive-Development Research and Mathematical Education.* Washington, DC: National Council of Teachers of Mathematics, 1971.

Nunley, I. M. McC. A Study of the Child's Ability to Understand the Concept of Function. Unpublished Doctoral dissertation, University of Oklahoma, 1973.

Orton, A. A Cross-Sectional Study of the Development of the Mathematical Concept of a Function in Secondary School Children of Average and Above Average Ability. Unpublished Master's thesis, University of Leeds, 1970.

Page, D. A. *Number Lines, Functions, and Fundamental Topics.* New York: Macmillan, 1964.

Sawyer, W. W. Prelude to Mathematics. Baltimore: Penguin Books, 1955.

Scandura, J. M. Mathematics: Concrete Behavioral Foundations. New York: Harper & Row, 1971.

Seymour, D. and Shedd, M. Finite Differences (A Pattern-discovery Approach to Problem Solving). Palo Alto, California: Creative Publications, 1973.

Thomas, H. L. The Concept of Function. In M. F. Rosskopf (Ed.), Children's Mathematical Concepts: Six Piagetian Studies in Mathematics Education. New York: Teachers College Press, 1975.

Wirtz, R. W. Nonverbal Instruction. Arithmetic Teacher 10: 72-77; 1963.

Wirtz, R. W. and Botel, M. Math Workshop for Children. Chicago: Encyclopedia Press, 1961.

Wirtz, R. W.; Botel, M.; and Nunley, B. G. Discovery in Elementary School Mathematics. Chicago: Encyclopedia Britannica Press, 1963.

Wittrock, M. C. A Generative Model of Mathematics Learning. Journal for Research in Mathematics Education 5: 181-196; 1974.

Footnote

[1]The essential features of a *function* have been identified clearly by Allendoerfer and Oakley (1963), among many others, who also describe it as "a special case of relation" which in turn "is a set of ordered pairs" (p. 195). More explicitly:

> A *function f* is a relationship between two sets: (1) a set *X* called the *domain of definition* and (2) a set *Y* called the *range*, or *set of values*, which is defined by (3) a rule that assigns to each element of *X* a unique element of *Y*. . . .

This definition may be more compactly stated as follows:

> A *function f* is a set of ordered pairs (x,y) where (1) x is an element of a set X, (2) y is an element of a set Y, and (3) no two pairs in f have the same first element. (p. 189)

Although the preceding definition (or the essence thereof) is commonly accepted and used, Buck (1970) has contended that

> Experience seems to show that the 'a function is a class of ordered pairs' approach is one which imposes severe limitations upon the student and provides a poor preparation for any further work with functions, either in school or later. (p. 255)

and has cited MacLane's plea that

> one should no longer preach that a function *is* a certain sort of set of ordered pairs. (Quoted by Buck, 1970, p. 255)

For purposes of this paper, it will be more suitable to adopt Buck's (1970) less formal characterization of function:

> A function on *A* to *B* . . . is a *rule* by which we assign to each member of set *A* some [unique] member of set *B*. (pp. 250-251, italics added)

It is more in keeping with the intent of this paper's suggested use of programmable calculators to focus upon rules than upon sets of ordered pairs.

APPENDIX

SIXRULE

Given prestored values of a and b (unknown to student):

Input x, calculate y for any one of the six rules indicated below.

Form of HP 41C's 82143A Peripheral Printer output: $x \rightarrow y$ for each rule.

RULE A. $y = ax + b$. □ → a□ $+ b$.

RULE B. $y = ax - b$. □ → a□ $- b$.

RULE C. $y = b - ax$. □ → $b - a$□.

RULE D. $y = (x + a)b$. □ → (□ $+ a)b$.

RULE E. $y = (x - a)b$. □ → (□ $- a)b$.

RULE F. $y = (a - x)b$. □ → $(a -$ □$)b$.

FOURULE

I. Input *counting* numbers x and y, calculate z for either of the two rules indicated below (A or B).

Form of HP-41C's 82143A Peripheral Printer output: $x, y \rightarrow z$ for each rule.

RULE A. z is the HCF of x and y.

RULE B. z is the LCM of x and y.

II. Input *counting* numbers x and y, calculate w and z for either of the two rules indicated below (C or D).

Form of HP-41C's 82143A Peripheral Printer output: $x, y \rightarrow w, z$ for each rule.

RULE C. w is the HCF of x and y, z is the LCM.

RULE D. w is the LCM of x and y, z is the HCF.

TRIRULE

Given prestored *counting* number d (unknown to student):

Input x, calculate y for any one of the three rules indicated below.

Form of HP-41C's 82143A Peripheral Printer output: $x \to y$ for each rule.

RULE A. y is the "rounded" (up or down) quotient when x is divided by d.*

RULE B. y is the "truncated" (always "rounded" down) quotient when x is divided by d; i.e., $y = [x/d]$, a whole number.

RULE C. y is the *remainder* when x is divided by d, such that $x = dq + y$ (q a whole number, $y < d$).

* y is a whole number.

PYTR

PYTR may be used by students to input pairs of positive integers m and n and generate corresponding triples of positive integers a, b, c that are Pythagorean--i.e., such that $a^2 + b^2 = c^2$.

Depending upon a given input, the resulting triple may be either:

Primitive--i.e., HCF $(a,b,c) = 1$; or

Not primitive--i.e., HCF $(a,b,c) > 1$.

Students generate data in an attempt to answer the following principal question solely on the basis of user input and calculator output, without knowing how output is generated from input:

WHAT INPUT CONDITION(S) MUST PREVAIL SO THAT m, n MAY BE SELECTED TO GENERATE A PRIMITIVE TRIPLE a,b,c?

The student sheets, pp. 69-73, are based upon a program written for the HP 41C calculator and accompanying 82143A Peripheral Printer.

(The program has been written to print explicit error messages in the case of *some but not all* forms of invalid or unsuitable input.)

[Student Record Sheet]

WHAT'S MY RULE?			#
	INPUT	OUTPUT	
A			
B			
C			
D			
E			
F			
G			
H			
I			
J			

THE RULE IS: ☐ ⟶

[Format for any SIXRULE problem.]

[Alternative student record sheet]

WHAT'S MY RULE? #

	Input	Rule?	Output
A			
B			
C			
D			
E			
F			
G			
H			
I			
J			

THE RULE IS:

PYTR: 1

The HP 41C calculator (with printer) has been programmed to generate mappings or assignments of the following form:

Input → Output

$(m,n) \rightarrow (a,b,c)$

where m, n, a, b, c are *POSITIVE INTEGERS* such that $a^2 + b^2 = c^2$, making (a,b,c) a *Pythagorean triple*. Some triples are termed *primitive*; others are not.

Use the information on the attached sheet [with relatively small positive integers (less than 100, for instance) for m and n] to generate mappings or assignments that will enable you to answer the following questions:

1. For a given $(m,n) \rightarrow (a,b,c)$, does (n,m) generate the same (a,b,c)?

2. Under what condition(s) will an input (m,n) be rejected and not generate any triple?

3. What distinguishes a *primitive* triple from one that is not primitive?

4. Under what condition(s) does (m,n) generate a *primitive* triple?

BONUS question:

What is the "rule" by which (m,n) generates a of (a,b,c)?

b of (a,b,c)?

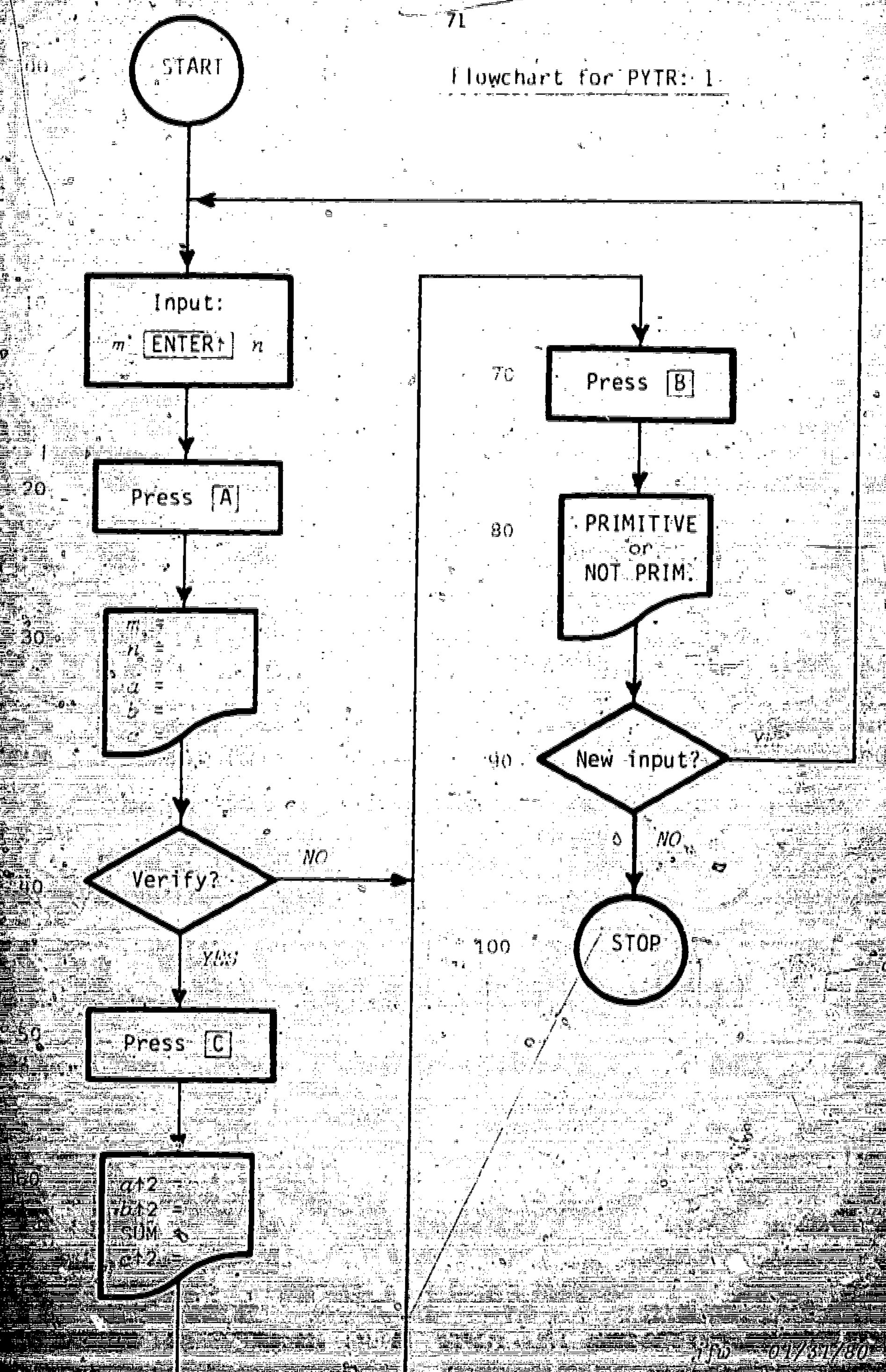
c of (a,b,c)?

Using PYTR-1

STEP	INSTRUCTION	DO THIS		PRINTER OUTPUT
1	Input m and n; calculate a, b, c:	m n	ENTER ↑ A	$m =$ $n =$ $a =$ $b =$ $c =$
2	OPTIONAL; verify that $a^2 + b^2 = c^2$: (a↑2 means a^2, b↑2 means b^2, SUM means $a^2 + b^2$, c↑2 means c^2)		C	a↑2 = b↑2 = SUM = c↑2 =
3	Determine whether (a,b,c) is a primitive triple: (Sometimes it takes several seconds for calculator to print output.)		B	PRIMITIVE or NOT PRIMITIVE
4	Return to Step 1 for next input.			

You may prefer to use the flowchart version (page 71) of these instructions.

Flowchart for PYTR: 1



jfw 01/31/80

Using PYTR: 2

Step 1. m [ENTER↑]

n [A] . . . . m =
n =

a =
b =
c =

*Step 2. [C] . . . . a↑2 =
b↑2 =
SUM =
c↑2 =

Step 3. [B] . . . . PRIMITIVE or NOT PRIMITIVE

If PRIMITIVE, go to 1 for new input if desired.
If NOT PRIMITIVE, continue:

Step 4. [D] . . . . CFCT =

a =
b =
c =

*Step 5. [C] . . . . a↑2 =
b↑2 =
SUM =
c↑2 =

Step 6. [B] . . . .

Return to Step 1 for new input if desired.

**OPTIONAL step.*

At step 4:

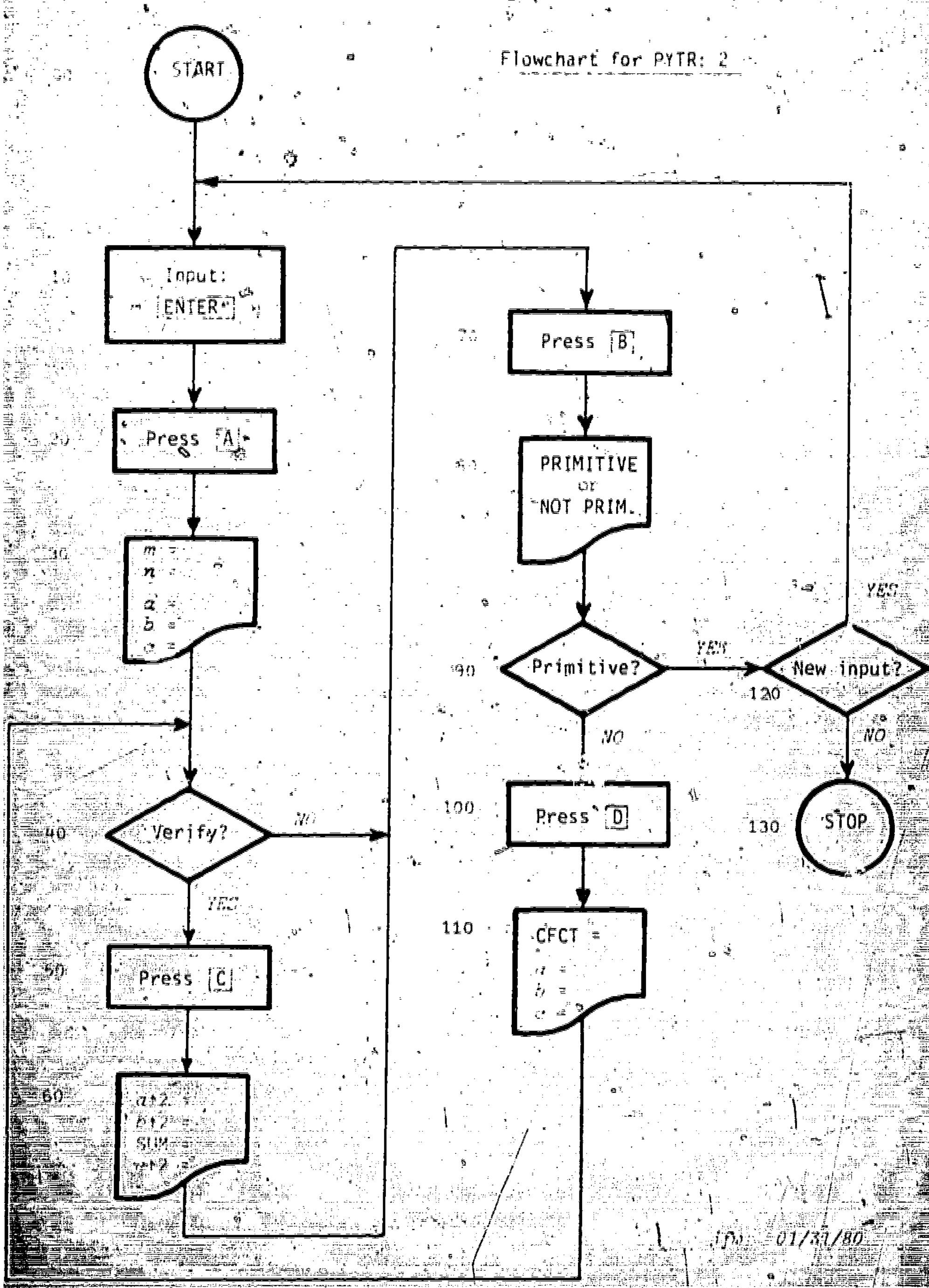
What does the value of CFCT represent?

How are a, b, c generated?

Will the Step 6 printed output always be the same? If so, why? If not, why not?

What would happen if the Step 3 output read PRIMITIVE, and you continued with 4, 5, 6?

Flowchart for PYTR: 2



IPD: 01/31/80

INPUT/OUTPUT Samples

1. 4 [ENTER↑]
   0 [A] — m = 4.
   n = 0. [BEEP]
   0 IS INVALID.
   TRY NEW INPUT

2. 0 [ENTER↑]
   6 [A] — m = 0.
   n = 6. [BEEP]
   0 IS INVALID.
   TRY NEW INPUT.

3. 6 [ENTER↑]
   9 [A] — m = 6.
   n = 9.
   a = 108.
   b = 45.
   c = 117.
   [B] — NOT PRIMITIVE

4. 8 [ENTER↑]
   5 [A] — m = 8.
   n = 5.
   a = 80.
   b = 39.
   c = 89.
   [B] — PRIMITIVE

5. 11 [ENTER↑]
   [A] — m = 11.
   n = 11. [BEEP]
   INVALID INPUT.
   TRY NEW INPUT
   USING m ≠ n.

6. 12 [ENTER↑]
   18 [A] — m = 12.
   n = 18.
   a = 432.
   b = 180.
   c = 468.
   [C] — a↑2 = 186,624.
   b↑2 = 32,400.
   SUM = 219,024.
   c↑2 = 219,024.
   [B] — NOT PRIMITIVE
   [D] — CFGT = 36.
   a = 12.
   b = 5.
   c = 13.
   [C] — a↑2 = 144.
   b↑2 = 25.
   SUM = 169.
   c↑2 = 169.
   [B] — PRIMITIVE

NOTES

SIXRULE:

The HP 41C program has been written to accommodate any one of the following options (with prestored a and b restricted accordingly):*

1. Input/Output are restricted to *non-negative integers*, with an explicit error message printed if this restriction is violated.

2. Input/Output are restricted to *integers* (negative as well as non-negative), with an explicit error message printed if this restriction is violated.

3. Input/Output are restricted to *non-negative rational numbers*, with an explicit error message printed if this restriction is violated.

4. Input/Output may be any rational number (within calculator range), negative as well as non-negative.

FOURUEE:

The HP 41C program has been written to print an explicit error message if input, x or y, is *not* a *counting* number.

TRIRULE:

The HP 41C program has been written to print an explicit error message if input a is *not* a *counting* number.

(The instructor-prestored d should also be a *counting* number; however, NO provision has been made in the program as written for an error message if this condition is violated.)

*Since these values are instructor-prestored, it is assumed that any applicable restrictions have been met and no error-message provision has been included in the program to cover "violations."

Illustrative error messages, each of which is preceded by an audible BEEP:

[illegible]ORRY, BUT, [illegible] WILL NOT WORK. [illegible] NEW INPUT.	SORRY, BUT, [illegible] WILL NOT WORK. TRY NEW INPUT.	SORRY, BUT 4 WILL NOT WORK. TRY NEW INPUT.	SORRY. USE ONLY COUNTING NUMBERS. TRY NEW INP[illegible]

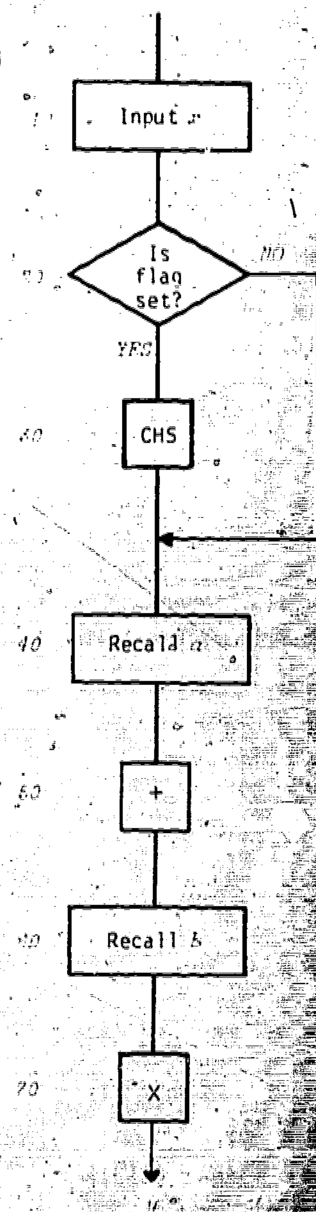
SIXRULE

Prestore/Initialize for RULE

D: *a* [ENTER↑] *b* [CF] §

E: *a* [CHS] [ENTER↑] *b* [CF] §

F: *a* [ENTER↑] *b* [SF] §

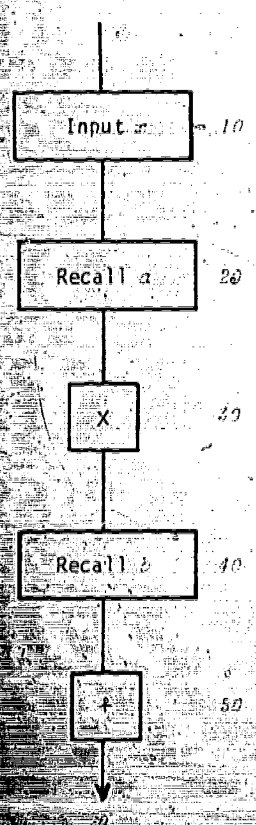


Prestore/Initialize for RULE

A: *a* [ENTER↑] *b* ∇

B: *a* [ENTER↑] *b* [CHS] ∇

C: *a* [CHS] [ENTER↑] *b* ∇



∇ Execute from keyboard.

§ Execute from keyboard, with flag setting "programmed"

PYTR Program for the HP-41C

```
01•LBL "PYTR"
02•LBL [illegible]
03 [illegible]
04 GTO 05
05•LBL E
06 SF 00
07•LBL 05
08 XEQ A
09 FS? 01
10 STOP
11 FS? 00
12 XEQ C
13 XEQ B
14 X=Y?
15 RTN
16 XEQ D
17 FS? 00
18 XEQ C
19 XEQ B
20 RTN
21•LBL D
22 ADV
23 SF 12
24 CF 13
25 "CFCT = "
26 ARCL X
27 AVIEW
28 ADV
29 SF 13
30 ST/ 03
31 ST/ 04
32 ST/ 05
33 ST/ 06
34 ST/ 07
35 RCL 03
36 RCL 04
37 RCL 05
38 GTO 01
39•LBL A
40 CF 01
41 SF 12
42 SF 13
43 ADV
44 ADV
45 STO 02
46 X<>Y
47 STO 01
48 "M = "
49 ARCL X
50 AVIEW
51 "N = "
52 ARCL Y
53 AVIEW
54 ADV
55 X=Y?
56 GTO 06
57 [illegible]
58 X=0?
59 GTO 07
[illegible]
[illegible]
62 STO 03
63 STO 06
64 RCL 01
65 X↑2
66 RCL 02
67 X↑2
68 -
69 ABS
70 STO 04
71 STO 07
72 RCL 01
73 X↑2
74 RCL 02
75 X↑2
76 +
77 STO 05
78•LBL 01
79 "A = "
80 ARCL Z
81 AVIEW
82 "B = "
83 ARCL Y
84 AVIEW
85 "C = "
86 ARCL X
87 AVIEW
88 RTN
89•LBL C
90 ADV
91 CF 12
92 SF 13
93 RCL 05
94 X↑2
95 RCL 04
96 X↑2
97 RCL 03
98 X↑2
99 "A↑2 = "
100 ARCL X
101 ACA
102 ADV
103 "B↑2 = "
104 ARCL Y
105 ACA
106 ADV
107 +
108 CF 13
109 "SUM = "
110 ARCL Y
111 ACA
112 ADV
113 SF 13
114 "C↑2 = "
115 ARCL Y
116 ACA
117 ADV
118 RTN
119•LBL B
120 ADV
121•LBL [illegible]
122 RCL 06
123 ENTER↑
124 ENTER↑
125 RCL 07
126 /
127 INT
128 RCL 07
129 STO 06
130 *
131 -
132 X=0?
133 GTO 03
134 STO 07
135 GTO 02
136•LBL 03
137 1
138 RCL 06
139 X=Y?
140 GTO 04
141 SF 12
142 CF 13
143 "NOT "
144 ACA
145 CF 12
146 "PRIMITIVE"
147 ACA
148 PRBUF
149 RTN
150•LBL 04
151 SF 12
152 CF 13
153 "PRIMITIVE"
154 ACA
155 PRBUF
156 RTN
157•LBL 06
158 SF 01
159 CF 12
160 CF 13
161 "INVALID INPUT,"
162 PRA
163 "TRY NEW INPUT,"
164 PRA
165 "USING "
166 ACA
167 SF 12
168 SF 13
169 BEEP
170 " M ≠ N."
171 ACA
172 PRBUF
173 RTN
174•LBL 07
175 SF 01
176 CF 12
177 CF 13
178 BEEP
179 "0 IS INVALID."
180 PRA
181 "TRY NEW INPUT,"
182 PRA
183 END
```

Additional PYTR Notes

1. The program is to be executed with the calculator in USER mode and the printer in MAN mode. No provision has been made for executing the program *without* the printer.

2. The program assumes that the user is "intellectually honest" and restricts input to *positive integers*, although an error message is printed if 0 is mistakenly (or otherwise!) used for m or n. In its present form, however, the program does *not* reject a *non*integral input for m or n, nor does it reject an integral input for which $m < 0$ or $n < 0$. (The program could be modified, of course, to reject such inputs also.)

3. a,b,c is computed from m,n as follows:

$$a = 2mn,$$
$$b = |m^2 - n^2|,$$
$$c = m^2 + n^2.$$

By using the absolute value of the difference between m^2 and n^2 to compute b, it is unnecessary to invoke the input condition that $m > n$. It is left for the student to "discover" that input order (m,n vs. n,m) has no effect upon the a,b,c triple computed (which is primitive iff m and n are *relatively prime* and of *opposite parity*).

4. In Step 3 (PYTR 1 or PYTR 2) the Euclidean algorithm is used to calculate the HCF of a and b for the a,b,c triple generated by Step 1,* and the value of the HCF (1 if a primitive triple) is *displayed but not printed or identified as such* as final calculator output when Step 3 terminates.

For PYTR 2, if HCF $\neq$ 1 its value is printed at the outset of Step 4 and identified as CFCT but its meaning remains to be "discovered" by users. The HCF is used as the divisor of a and b and c of the Step 1 nonprimitive triple to generate the primitive triple printed in Step 4.

5. Two shortcuts have been programmed in connection with PYTR 2:

(1). m [ENTER↑]

n [E] Executes [A] [C] [B] as subroutines, and if necessary executes [D] [C] [B] to generate a primitive triple.

(2). m [ENTER↑]

n [J] Same as (1), except [C] is not executed.

*The HCF of a,b is also necessarily the HCF of a,b,c.

6. Data storage: $m \rightarrow R_{01}$

$n \rightarrow R_{02}$

$a \rightarrow R_{03}$ and R_{06}

$b \rightarrow R_{04}$ and R_{07}

$c \rightarrow R_{05}$

7. Flags 12 and 13 are used to control printout format.

Flag 00 is used to determine whether [C] is included as a subroutine of [E] or excluded as a subroutine of [J]. Flag 01 is related to this also.

References

Archambeau, Sister M. L. Pythagorean Triples Grouped into Families. Mathematics Teacher 61: 251-252; 1968.

Barnett, I. A. Some Ideas About Number Theory. Washington: National Council of Teachers of Mathematics, 1961, pp. 48-49.

Gardner, M. Sixth Book of Mathematical Games for Scientific American. New York: Scribner, 1971, pp. 158-160.

Glenn, W. H. and Johnson, Donovan. The Pythagorean Theorem (Exploring Mathematics on Your Own). Manchester, Missouri: Webster, 1960, pp. 9-12.

Ogilvy, C. S. and Anderson, J. T. Excursions in Number Theory. New York: Oxford University Press, 1966, pp. 65-67.

Ore, O. Invitation to Number Theory (Vol. 20, New Mathematics Library). New York: Random House, 1967, pp. 49-53.

Roensch, S. Finding Pythagorean Triples. Wisconsin Teacher of Mathematics 28(1): 1977.

Rothbart, A. and Paulsell, B. Pythagorean Triples: A New, Easy-to-Derive Formula with Some Geometric Applications. Mathematics Teacher 67: 215-218; 1974.

Watson, J. D. and Comella, J. J. Pythagorean Triples: What Kind? How Many? Mathematics Teacher 69: 108-110; 1976.

Young, F. H. Pythagorean Numbers; Congruences, A Finite Arithmetic; Geometry in the Number Plane. Lexington, Massachusetts: Ginn, 1961, pp. 1-5.

SOME HIGHLIGHTS FROM THE HISTORY OF PI ON THE PROGRAMMABLE CALCULATOR

Eli Maor
Department of Mathematics
University of Wisconsin-Eau Claire
Eau Claire, Wisconsin 54701

The history of mathematics offers an almost endless source of enrichment material with which to enlighten a classroom discussion. Nowadays, with the programmable hand-held calculator having become so inexpensive, one can illustrate many problems on one's own instrument and run them at home, in the classroom, or even on a vacation. Let me discuss here some highlights from the history of the number π. Of course, for a more detailed discussion, one should consult any of the many sources on the history of mathematics, some of which are mentioned in the bibliographical list.

The number π has intrigued scholars and laymen alike since the dawn of recorded history. The famous Rhind papyrus (ca. 1650 B.C.) uses the approximation $\pi \sim (4/3)^4 = 3.1604938...$, which is within 0.6 percent of the exact value. Another interesting approximation to π is the easily remembered fraction $355/113 = 3.1415929...$, discovered by the Chinese Tsu Ch'ung-chih around 480 A.D. It is surprising that the Biblical value for π is simply 3, as is clear from a statement in I Kings vii 23: "And he made a molten sea, ten cubits from the one brim to the other; it was round all about, and his height was five cubits: and a line of thirty cubits did compass it round about." Thus the ancient Egyptians had already been using a much better approximation some fifteen centuries before.

But while all the ancient values were based on an actual measurement of the circumference-to-diameter ratio for given circles, it was the Greeks who first proposed an algorithm--that is, a systematic procedure--to find π to any desired accuracy. This was the famous method of exhaustion, invented by Archimedes of Syracuse (ca. 287 - 212 B.C.). By inscribing and circumscribing regular polygons of an increasing number of sides n around a circle of radius R, he showed that the value of π is "squeezed" between the values $n \cdot \sin(180°/n)$ and $n \cdot \tan(180°/n)$ for any given n. (Of course, he did not use the modern trigonometric notation, but the formulas are essentially his.) He began with an equilateral triangle (n = 3) and then doubled n five

---

Eli Maor teaches undergraduate and graduate mathematics courses. He has special interests in applied mathematics, mathematics education, and the history of mathematics. In 1978 he developed programmable calculator courses for gifted elementary and junior high school children.

times up to n = 96, for which π is squeezed between 3.1410320... and 3.1427146... . It is an easy task to write a program that will display these values for n = 3, 6, 12, ... (Do not forget to put your calculator in the "degree" mode!)

All subsequent methods of approximating π were essentially variations of the exhaustion method. It was not until 1579 that the French mathematician Francois Viète gave a new method based on an <u>infinite product</u>:

$$\frac{1}{\pi} = \frac{1}{2} \cdot \frac{\sqrt{2}}{2} \cdot \frac{\sqrt{2+\sqrt{2}}}{2} \cdot \frac{\sqrt{2+\sqrt{2+\sqrt{2}}}}{2} \cdot \ldots$$

This remarkable formula shows that π can be calculated solely from the number 2 by a succession of additions, multiplications, divisions, and square-root extractions. Once again, it is interesting to write a program which will approximate π from this formula, using a partial products. A program for the Texas Instruments TI 5. follows:

LRN			
2	RCL 1	RCL 1	LRN
1/x	$\sqrt{x}$	÷	RST
STO 0	STO 1	2	R/S
Lbl 0	RCL 0	=	
2	1/x	Prd 0	(18 program steps)
SUM 1	Pause	GTO 0	

It becomes a fascinating experience to watch the numbers in the display as they gradually approach π. The convergence is very fast, and after only 15 partial products, the displayed value is correct to seven decimal places.

Another remarkable product leading to π was discovered by the English mathematician John Wallis in 1650 and is named after him:

$$\frac{\pi}{2} = \frac{2 \cdot 2 \cdot 4 \cdot 4 \cdot 6 \cdot 6 \cdot 8 \ldots}{1 \cdot 3 \cdot 3 \cdot 5 \cdot 5 \cdot 7 \cdot 7 \ldots}$$

Again it is easy and instructive to write a program which will approximate π from the partial products of this formula. The details are left to the reader.

Let me now mention some infinite series which involve π, many of which mark milestones in the history of mathematics. The first such series was discovered in 1671 by the Scotch James Gregory from the power series for $\tan^{-1}x$:

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + - \ldots$$

(The series is also known as the Gregory-Leibniz series.) It was one of the first applications of the newly invented differential and integral calculus, even though it is quite useless as a practical means to calculate π, due to its slow convergence. In writing a program to approximate π from the partial sums of this series, one has to take into account the alternating signs of the terms. This can be done by storing (-1) in some memory and then instruct the instrument to multiply the content of this memory by (-1) at every execution of the loop.

It is well known that the harmonic series--the sum of the reciprocals of the natural numbers--diverges. However, for many years it was not known if the corresponding series with the squares of the natural numbers diverges or converges, and if it converges, to what limit. This intriguing question was solved in 1736 by Leonhard Euler, who showed that the series converges to $\pi^2/6$:

$$1/1^2 + 1/2^2 + 1/3^3 + 1/4^2 + \ldots = \pi^2/6.$$

It is always fascinating--mystifying, indeed--to discover such a remarkable relation between the natural numbers and π, which is transcendental. A TI 57 program to approximate π from this series follows:

LRN

1	1/x	6	Pause
SUM 0	SUM 1	=	RST
RCL 0	RCL 1	$\sqrt{x}$	
x^2	x		LRN
			RST
			R/S

One cannot only use the programmable calculator to approximate π from these series, but also to compare their rates of convergence: one only has to halt the program at any desired stage by pressing R/S, then RCL 0, and the number of partial sums will be displayed. Pressing R/S again will resume execution of the program.

It turns out that both the Gregory series and the Euler series converge very slowly, taking 628 and 600 terms respectively to find π to two decimal places (i.e., π = 3.14). For the Gregory series, of course, the convergence will be oscillating, approaching alternatingly from above and below.

Some other "generalized harmonic series" (i.e., series involving the reciprocals of powers of the natural numbers) also involve π, such as the fast-converging series

$$\frac{\pi^4}{90} = \frac{1}{1^4} + \frac{1}{2^4} + \frac{1}{3^4} + \frac{1}{4^4} + \ldots$$

Many more series of this kind are known and can be derived either from the Taylor or Fourier expansion of various elementary functions. For details, see Courant (1956).

References

Beckmann, Petr. A History of π. Boulder, Colorado: Golem Press, 1971.

Courant, Richard. Differential and Integral Calculus, Volumes I and II. London: Blackie & Sons, 1956.

Eves, Howard. An Introduction to the History of Mathematics (4th ed.). New York: Holt, Rinehart and Winston, 1976.

[illegible]

[illegible]

[illegible] preprogrammed [illegible]. Prerequisite is knowledge of [illegible] Algebra. This is a one-quarter-hour credit course with twelve class meetings per quarter and a Pass-No Pass grade based upon satisfactory completion of assignments. We meet twice a week for six weeks during the first part of the quarter. This prepares students to use their calculators in other classes early in the quarter.

Students can enroll in either of two types of sections: AL[illegible] or RPN (Reverse Polish Notation). In the ALG sections [illegible] RPN [illegible] the same type of calculator by all students [illegible] provides uniformity of instruction. [illegible] Some [illegible] more expensive [illegible] or even HP [illegible] have no difficulty using the instructional materials provided for the class; however, the Department of Mathematics has available approximately ten each of the TI-[illegible], TI-[illegible], and HP 33E calculators for student use [illegible] students can take a [illegible] calculator to use on homework assignments.

The first four or five sessions are devoted to becoming acquainted with the nonprogramming features of the calculators; we assume no previous experience. During the first two weeks, the groups are not very uniform in terms of calculator background; however, by the time we begin programming, the entire group is at about the same level.

Instructional materials are specifically designed for this course and are made available to the students in dittoed form. These materials are viewed as experimental; we expect to make revisions based on experiences of the instructors during the coming school year. We have 12 lessons, each of which has a specific objective. For instance, each of the first four lessons is intended to have the student become acquainted with the efficient use of specific keys in nonprogramming mode. After the student understands how the calculator works in keyboard mode, programming is introduced. The

---

[illegible] has directed numerous NSF summer institutes in mathematics for secondary and junior college teachers. He is currently interested in curriculum development, particularly for precalculus mathematics, and he continues to design and teach calculator courses for college students.

[illegible]

[illegible]

[illegible]

[illegible] contribute to the mathe- [illegible] in evaluating polynomials by using the [y^x] key, difficulties occur when evaluating [illegible] negative number. To overcome this we introduce the idea of nested form. This also allows us to discuss "synthetic division by calculator" (see Section 9 of Syllabus and Problem 5 of Sample Problems).

The material in each lesson consists of a brief description (intended to be supplemented by the instructor) of the topic, followed by a set of problems to be assigned for homework. A list of topics is included in the Syllabus.

[illegible] the RPN Sections

In order to make efficient use of RPN calculators it is necessary that students understand use of the stack. Thus we begin the course with practice in using the stack and we encourage students to keep track of stack contents particularly in problems involving programming (even though we do not explicitly ask for it in problem sets).

As with the AlG sections we introduce the idea of nested form for evaluation of polynomials even though the [y^x] key of the HP-33E will accept negative base numbers when the exponent is an integer. Nested form is also related to synthetic division. The Syllabus includes the list of topics in the RPN sections.

[illegible]

[illegible]

1. [illegible]

2. [illegible]

3. [illegible]

4. [illegible]

5. [illegible]

6. Introduction to programming; evaluation of [illegible] functions.

7. Programming quadratic formula; general quadratic, cubic and quadratic functions; rational functions.

---

8. Becoming acquainted with the TI 58; data storage registers, addresses; nonprogramming problems.

9. Programming evaluation of polynomials in nested form; synthetic division by calculator.

10. Using label keys; evaluating functions for equally incremented values of the independent variable; editing programs.

11. Programming involving unconditional transfer: [GTO], [SBR]; evaluating sum, difference, product, quotient, or composition of functions.

12. Programming involving conditional transfer; t-register; [illegible]

1. [illegible]

2. [illegible]

3. [illegible]

4. [illegible]

5. [illegible]

6. Intro[illegible] evaluating polynomials and rational [illegible]

7. [illegible] functions [illegible] implemented [illegible]

8. [illegible] differences, products, quotients, or compositions [illegible] terms.

9. [illegible] evaluate polynomials; synthetic division by [illegible]

10. [illegible] unconditional transfer: GTO, SBR; deleting and insert[illegible] in programs.

11. [illegible] conditional transfer.

12. [illegible] determine limiting values; sequences; intro- [illegible]

[illegible]

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	2.0
[illegible]									

[illegible] starting value [illegible] increment [illegible] final value of x = [illegible].

5. Using [illegible] as it relates to synthetic division, program [illegible] the quotient and remainder when [illegible] is divided by

(a) [illegible] (b) [illegible] (c) [illegible].

6. [illegible] the compound interest formula

$$A = [illegible]\left(1 + \frac{[illegible]}{m}\right)^{mn}$$

for various values of m and n. Determine the value of A for n = [illegible] years when interest is compounded annually (m = 1); quarterly (m = 4); monthly (m = 12); daily (m = 365).

7. Suppose $f(x) = 3x^2 - [illegible]x - [illegible]$ and $g(x) = \sqrt{\frac{x+3}{x+1}}$.

Using [illegible], program your calculator to help you complete the following table. Round off entries to two decimal places. In case the calculator gives an "Error" response, explain why it does.

[illegible]

8. [illegible] general quadratic function, do [illegible]

9. [illegible] evaluation [illegible]

[illegible] 0.1; 0.01; 0.001; [illegible]

−.1; −.01; −.001; −0.0001; [illegible]

[illegible] your program the feature of automatically determining [illegible]

10. Program the evaluation of the sequence

a_0 = any positive integer

$$a_{n+1} = \begin{cases} \frac{a_n}{2} & \text{if } a_n \text{ is even} \\ 3a_n+1 & \text{if } a_n \text{ is odd} \end{cases}$$

Find sequences for several different starting values of a_0. In each case get enough terms until you observe something [illegible] happening.

11. Suppose $S_n = \sum_{k=0}^{n} \frac{1}{k!}$. Determine S_n for $n = 1,2,3,\ldots,10$.

12. Fixed point problems. Suppose we wish to find the roots of $f(x) - x = 0$. Let x_0 be an initial guess; then evaluate $f(x_0)$; $f(f(x_0))$; $f(f(f(x_0)))$, ...

Try this on:

(a) $\cos x - x = 0$ (b) $\sin x - x^3 = 0$. (Use $\sqrt[3]{\sin x} = x$)

(c) $e^{-x} - x = 0$ (d) $x^3 - 2x^2 + 5x - 3 = 0$. (Try $\frac{3}{x^2-2x+5} = x$).

[illegible]

Aronofsky, Julius S.; Frame, Robert J.; and Greynolds, Elbert B., Jr. Programmable Calculators: Business Applications. New York: McGraw-Hill, 1978.

This book stresses basic computational skills and elementary and advanced programming techniques, particularly for Texas Instruments programmable calculators.

Ball, John A. Algorithms for RPN Calculators. New York: Wiley, 1977.

Barnes, John E. and Waring, Alan J. Pocket Programmable Calculators in Biochemistry. New York: Wiley, 1979.

[illegible] Introduction to Differential Equations. Boston: Houghton-Mifflin, [illegible].

Jill Coupe is a reference librarian. She has been an abstractor and indexer for the National Clearinghouse for Mental Health Information. Her interests include writing short stories and novels.

*Annotations of ERIC documents have been adapted from original ERIC abstracts.

[illegible]

[illegible]

[illegible] Stephen L. [illegible] and Mark A. [illegible]

Eisberg, Robert M. Applied Mathematical Physics with Programmable [illegible]

[illegible] as a supplement to existing physics texts for the [illegible] programmable [illegible] capabilities presents a series of programs that [illegible] to treat key topics in physics.

Eisberg, Robert M. and Hyde, [illegible]. [illegible]: Skydiver, Rocket, and Satellite Motion on Programmable Calculators. Portland, OR: [illegible] Press, [illegible].

Engelsohn, Harold S. Programming Programmable Calculators. Rochelle Park, NJ: Hayden, [illegible].

Etlinger, Leonard E.; Krull, Sarah; Sachs, Jerry; and Stolarz, Theodore J. The Calculator in the Classroom: Revolution or Revelation? Chicago: Chicago State University, 1980.

Advantages of the programmable calculator and recommendations for its use in the secondary school classroom are presented.

Henrici, Peter. Computational Analysis with the HP-25 Pocket Calculator. New York: Wiley, 1977.

Thirty high-level mathematical programs written for the HP-25 programmable calculator are given. By means of the flow diagrams and the detailed descriptions provided, the programs are adaptable for any calculator of comparable capability.

Hewlett-Packard. The Programming Book. Knoxville, TN: Approach 13-5 Corporation, 1976.

This booklet covers steps in writing and using effective programs, special programming features, and consumer information on selecting a programmable calculator.

Hewlett-Packard. Games Pac I. Palo Alto, CA: Hewlett-Packard, 1977.

A collection of games for the HP 67 and HP 97 is presented. There are 19 programs, prerecorded on magnetic cards and varied in difficulty, which can be played by two or more people.

[illegible]

[illegible], Gerald [illegible]; Betty [illegible]; and Koesch, Carl. *Using Calculators in Mathematics 11.* Buffalo, NY: State University of New York, [illegible].

[illegible], Gerald [illegible]; Betty J.; Koesch, Carl; and [illegible], [illegible]. *Using Calculators in Mathematics 12.* Buffalo, NY: [illegible].

[illegible] [illegible], Carl. *Pocket Calculator Supplement for [illegible].* [illegible], MA: Addison-Wesley, 1979.

[illegible] to be used as a supplement to the teaching of calculus with any textbook, this book is particularly coordinated with *[illegible] and Analytic Geometry* ([illegible] edition) by George B. Thomas, Jr. and Ross L. Finney. Programs for the HP 33E and the TI 57 calculators are included.

[illegible], [illegible] and [illegible], [illegible] A. *Micro-computer [illegible] and [illegible].* [illegible], IL: Matrix, 1975.

[illegible] programmable calculator terms are included.

[illegible] *[illegible] and Dictionary.* Champaign, IL: [illegible], 1976.

[illegible] sections on what is available (1976) in programmable calculators and how to use most units in the $50 to $3000 price range are included, plus a 7000-term calculator dictionary.

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible] papers was presented at a symposium at the [illegible] NCTM Meeting, April 19[illegible].

[illegible] Stephen L. [illegible] Algebra, Puzzles and [illegible] Programmable Calculators and Personal Computers. [illegible] Prentice-Hall, 1981.

A collection of interesting problems and activities for solution by students, hobbyists, etc. All problems can be solved on programmable calculators. Hints and flowcharts are given. Answers [illegible] BASIC programs for [illegible] computers.

[illegible] The Programmable Calculator in the Classroom. [illegible] Ohio State University, 1980.

Teaches the skills necessary to develop algorithms for programmable calculators in the classroom.

Suydam, Marilyn N. Calculators: A Categorized Compilation of References. Columbus, OH: ERIC/SMEAC, 1979.

[illegible] references [illegible] most of them with annotations. Nearly [illegible] mention programmable calculators. This book was consulted in preparation of the present bibliography.

Texas Instruments. Fundamentals of Programming Using Hand-Held Calculators. Dallas: Texas Instruments, 197[illegible].

Developed by the University of Denver Mathematics Laboratory, this is a set of [illegible] masters [illegible] for teaching programming using hand-held programmable calculators.

[illegible]

[illegible] ... 1979.

[illegible]

[illegible] Calculator Information [illegible]

[illegible] Approximately [illegible] ... were consulted during the preparation of the present bibliography.

[illegible], Marilyn N. Calculators: A Categorized Compilation of References. [illegible] ERIC: ED [illegible] (148 pages)

References on calculators are listed, most of them with annotations. This document was consulted in preparation of the present bibliography.

[illegible] for the Calculating Amateur. Sky and Telescope [illegible]; May [illegible].

[illegible] are listed for the amateur astronomer who uses a programmable calculator.

[illegible]

Ma[illegible], [illegible]. A Summer Course with the TI 57 Programmable Calculator. Mathematics Teacher 73: 99-106; February 1980.

A [illegible] with a programmable calculator is described. The course was part of a summer enrichment program for upper elementary and junior high school students.

[illegible], Gary L. Using Programmable Calculators to ENLARGE the Problem Solving Skills of 10-12 Year Olds. Computing Teacher 8: 38-41; September 1980.

Suggestions are made for using programming as a vehicle for teaching problem-solving skills to fourth and fifth graders.

Wavrik, John J. Programmable Calculators for Elementary School Students. Calculators/Computers Magazine : 46-47; September-October 1978.

This article is the first in a series on the HP 25 programmable calculator. Basic operations are presented.

Wavrik, John J. Programmable Calculators for Elementary Schools. Calculators/Computers Magazine 2: 53-55; November/December 1978.

A lesson on simple programming for students in grades 4-6 is given. Several problems and examples are given, including temperature conversion, estimation, and number operations.

Wavrik, John J. A Short Presentation in "Computer Literacy" using Programmable Calculators. Calculators/Computers Magazine 2: 9-11; November/December 1978.

The programmable calculator is suggested as a means of introducing elementary school students to certain features of computers. A one-hour lesson for grades 5 and 6 using the HP 25 programmable calculator is presented, in which students are given experience entering and running programs.

Wavrik, John J. Programmable Calculators for Elementary School Students. Computing Teacher 6: 39-41; May 1979.

Two units of study are given for teaching elementary school students the use of programmable calculators. Storage registers and the program memory are considered.

C. Secondary School

Secondary School Agriculture (Vocational)

Trede, Larry D. Using a Programmable Calculator in Vo-Ag. Agricultural Education 52: 17-18; April 1980.

The capabilities of the programmable calculator and its possible uses by a vocational agriculture teacher are discussed.

Secondary School Biology

Vail, Roy. Programmable Calculators in Biology Classes. American Biology Teacher 36: 496-498; November 1974.

Several uses of programmable calculators in biology classes are mentioned, including charting exponential population curves, evolution by natural selection, and random genetic drift.

Secondary School Chemistry

Ehrlich, Amos. Programmable Calculator and Kinetics of Chemical Reactions. International Journal of Mathematical Education in Science and Technology 11: 385-389; July/September 1980.

Simulation techniques using a programmable calculator in the study of chemical reactions are presented.

Weldert, R. W. Programmable Pocket Electronic Calculators in the Classroom. Journal of Chemical Education 54: 628; October 1977.

The uses of programmable calculators in high school chemistry classes are discussed, including grading, laboratory exercises, computing T-scores, and a quantitative approach to chemical equilibrium.

Secondary School Mathematics

Andersen, Lyle et al. Making Comparisons: Ratios. Topical Module for Use in a Mathematics Laboratory Setting. 1973. ERIC: ED 183 509. (39 pages)

The objectives of this module on making comparisons and ratios include using ratios to compare sets of objects and expressing ratios as decimals or fractions in lowest terms. Six experiments are provided, plus directions for utilizing a programmable calculator or computer.

Bawtree, Michael. Numerical Solutions Without Calculus. Mathematics in School 8: 19-20; March 1979.

The programmable calculator can be used to obtain numerical solutions to equations. The program is given and the method illustrated.

Dennis, J. Richard and Thomas, David. Low-budget Computer Programming in Your School (An Alternative to the Cost of Large Computers). Illinois Series on Educational Applications of Computers, No. 14. 1976. ERIC: ED 138 291. (6 pages)

The programmable calculator can be used in teaching the concepts and the rudiments of computer programming and in computer problem solving. Twenty-five programming activities related to high school mathematics are listed.

Durapau, V. J. and Bernard, John. From Games to Mathematical Concepts via the Hand-held Programmable Calculator. International Journal of Mathematical Education in Science and Technology 10: 417-424; July/September 1979.

A few games are suggested for programmable calculators which can create an environment in which mathematical concepts are more easily formed.

Hoffman, Ronit. Teaching Mathematics from an Algorithmic Point of View with the Use of Pocket Calculator. Unpublished M.S. thesis, Tel Aviv University, 1977.

A unit for teaching algorithms and the use of the SR 56 programmable pocket calculator was developed for teachers of science or mathematics in high schools or comprehensive schools.

Kastner, Sheldon B. Remedial Mathematics Skills Program for Optional Assignment Pupils; School Year 1974-75. (New York City Board of Education Function No. 09-59678). 1975. ERIC ED 137 477. (20 pages)

This is an evaluation of a New York City School District educational project, the major objective of which was to increase student competency in math computational skills. Math labs equipped with calculators, printing calculators, and programmable calculators were available for student use. Program participants, on the average, made one-year gains in actual achievement.

Krist, Betty J. The Programmable Calculator in Senior High School: A Didactical Analysis. Unpublished Doctoral dissertation, State University of New York at Buffalo, 1980.

Krist, Betty J. Uses of Calculators in Secondary Mathematics. Columbus, OH: Calculator Information Center, Information Bulletin No. 8, September 1980.

LaBar, Martin; Wilcox, Floyd; and Richman, Claude M. Programmable Calculators as Teaching Aids and Alternatives to Computers. School Science and Mathematics 74: 647-650; December 1974.

The authors provide a list of calculators which have a capacity for handling programs, and a list of programs for such calculators which are available at cost. They argue that the use of these materials at many levels of mathematics instruction enhances both motivation and understanding.

Mattei, K. C. Courses about Computers--for Secondary School Students. Australian Mathematics Teacher 30: 118-121; June 1974.

A method of teaching introductory ideas about computer operations by using a programmable calculator is suggested.

Quinn, Donald Ray. The Effect of the Usage of a Programmable Calculator upon Achievement and Attitude of Eighth and Ninth Grade Algebra Students. Unpublished Doctoral dissertation, St. Louis University, 1975.

The programmable calculator was used in eighth- and ninth-grade algebra classes. When compared with students in noncalculator algebra classes, students using calculators showed less "anxiety toward mathematics" and had better "self-concept in mathematics," but no difference in achievement.

Sigurdson, Orville et al. Area. Topical Module for Use in a Mathematics Laboratory Setting. 1973. ERIC: ED 183 405. (61 pages)

This area package emphasizes three facets: the concept of area as a covering, the square unit, and formula development. Two enrichment activities are included, the first of which requires the aid of a programmable calculator or computer.

Snover, Stephen L. and Spikell, Mark A. The Role of Programmable Calculators and Computers in Mathematical Proofs. Mathematics Teacher 71: 745-750; December 1978.

This article illustrates the role of programmable calculators and computers in the creation of mathematical proofs by exploring a simple problem from number theory.

Snover, Stephen L. and Spikell, Mark A. Generally, How Do You Solve Equations? Mathematics Teacher 72: 326-336; May 1979.

Iterative techniques are presented for solving difficult equations with numerical methods that can be used easily on programmable calculators. Flowcharts and programs are given for TI 57, HP 25, and BASIC.

Snover, Stephen L. and Spikell, Mark A. $x = \frac{2}{x} + 1$, A Programmable Calculator Activity. New Jersey Mathematics Teacher 37: 6-8; Fall 1979.

Programs are given for the TI 57 and the HP 33E programmable calculators for one nonstandard problem.

Snover, Stephen L. and Spikell, Mark A. A Programmable Calculator Activity, $x = \frac{1}{x} + 1$. 1979. ERIC: ED 170 117. (7 pages)

A nonstandard activity which could not be easily explored without the use of a programmable calculator is presented, and flowcharts and programs for different programmable calculators are given.

Snover, Stephen L. and Spikell, Mark A. Programmable Calculators Facilitate Simple Solutions to Mathematical Problems. 1979. ERIC: ED 170 115. (8 pages)

Many types of problems ordinarily requiring advanced techniques or special insight to solve can now be done as simple programming exercises on inexpensive programmable calculators. The following examples are given: evaluating polynomials, finding limits, evaluating finite and infinite series, computing variable length products, searching for data, and developing proof.

Snover, Stephen L. and Spikell, Mark A. Using Programmable Calculators to Evaluate Complicated Formulas. 1979. ERIC: ED 170 116. (9 pages). Also in: Virginia Mathematics Teacher 6: 25-28; February 1980.

The use of the programmable calculator in evaluating complicated formulas is illustrated by considering the formula for finding

the area of any triangle when only the lengths of the three sides are known. Flowcharts and programs are given for the TI 57 and the HP 33E programmable calculators.

Secondary School Physics

Beare, Richard and New, Peter J. Programmable Calculators for Elementary Physics Teaching. Physics Education 12: 424-426; November 1977.

Operating characteristics and features of programmable hand-held calculators are compared.

Beare, Richard. Programmable Calculators, Part II: Their Use in Applying Simple Laws in Physics to Some Complex Problems. School Science Review 59: 269-284; December 1977. (Part I of this article is listed below under Secondary School Science.)

The use of programmable calculators to solve complex physics problems is described.

Reiland, Robert J. A Realistic Model Rocket Program for a Small Programmable Calculator. Calculators/Computers Magazine 2: 72-74; September/October 1978.

A program for a programmable calculator is given which predicts the altitude achievable by a model rocket.

Smith, Clifton L. Computing in Secondary Physics at Armdale, W.A. Australian Science Teachers Journal 22: 33-40; May 1976.

An Australian secondary school physics course utilizing an electronic programmable calculator and computer is described. Calculation techniques and functions, programming techniques, and simulation of physical systems are detailed. A summary of student responses to the program is included.

Summers, M. K. Use of a Programmable Pocket Calculator in A-level Physics Courses. School Science Review 60: 316-325; December 1978.

Secondary School Science

Beare, Richard. Programmable Calculators, Part I: Their Use in Teaching Science. School Science Review 59: 36-48; September 1977.

Advanced secondary school science exercises which are amenable to the use of a programmable calculator are given, and use of the calculator is compared to use of a computer terminal.

Craig, James C. Simulating Air Quality Investigations with the Programmable Calculator. Science Teacher 41: 38-42; April 1974.

Ways of using a programmable calculator to obtain air pollution data in a simulated experiment are described.

D. College and Postgraduate

College Biology

Blumenberg, Bennett and Spikell, Mark A. Calculator Programs in General Genetics: II. Nei's Indices of Genetic Distance, Protein Identity, Migration Rate, and Divergence Time. Journal of Heredity 69: 278-280; July/August 1978.

Two programs of general interest to the population geneticist are presented, written in algebraic notation for Texas Instruments calculators. These programs are written for a genetic system composed of five diallelic loci, but may be modified to accommodate loci comprised of more than two alleles.

Blumenberg, Bennett and Spikell, Mark A. Calculator Programs in General Genetics: III. Latter's Indices of Heterozygosity, Population Differentiation, and Genetic Distance. Journal of Heredity 71: 293-294; July/August 1980.

A program written in algebraic notation for Texas Instruments calculators, and intended for a genetic system embracing an infinite number of diallelic loci, is presented. This program may also be modified to accommodate loci some or all of which consist of more than two alleles.

Forbes, M...on L. Simulation of Natural Selection on the Programmable Calculator. Journal of College Science Teaching 8: 95-96; November 1978.

A model or game is described which enables students to experiment with equilibria and to trace rapidly gene frequency changes through time under postulated conditions of selection.

Spain, J. D. Teaching Basic Biological Simulation Techniques with the Programmable Calculator. 1972. ERIC: ED 079 990. (2 pages)

An introductory course on digital computer simulation in Biology, taught at Michigan Technological University using the Olivetti programmable 101 calculator, is discussed.

Spikell, Mark A. and Blumenberg, Bennett. Calculator Programs in General Genetics: I. Computing Genetic Distance. Journal of Heredity 68: 187-190; May/June 1977.

A program written for Texas Instruments calculators is presented which can be used for computing genetic difference and genetic distance from data from five diallelic loci. The program can easily be expanded to include any number of loci.

College Chemistry

Attard, Alfred E. and Lee, Henry C. X-ray Crystallographic Computations Using a Programmable Calculator. Journal of Chemical Education 56: 650; October 1979.

Six crystallographic programs developed to illustrate the range of usefulness of programmable calculators in chemical analysis are described. The programs are suitable for the laboratory analysis of X-ray diffraction data.

Brabson, G. Dana and Seegmiller, David W. Programmable Calculators Add a New Dimension to Laboratories. Journal of Chemical Education 47: 117-119; February 1970.

Uses of programmable calculators in college chemistry classes are discussed, and a specific example is given: study of the homogeneous N_2O_4 - NO_2 equilibrium.

Clark, C. J.; Kuemmerle, E. W.; and Lieto, L. R. Programmable Calculators: Uses in Freshman Chemistry Laboratories. Journal of Chemical Education 52: 423; July 1975.

Two uses of the programmable calculator in the freshman chemistry laboratory are suggested: to determine whether or not a student's raw data fall within acceptable tolerance limits, and to check the reliability of unknowns and grading of quantitative experiments.

Hayman, H.J.G. Stereoscopic Diagrams Prepared by a Desk Calculator and Plotter. Journal of Chemical Education 54: 31-34; January 1977.

The use of a Hewlett-Packard 9810A programmable calculator with plotter for drawing ball-and-line stereopairs as well as three-dimensional structural formulas which are useful for teaching stereochemical principles and molecular structure is discussed.

Holdsworth, David. Applications of Programmable Calculators in Chemistry Classes. Australian Science Teachers Journal 23: 74-76; May 1977.

Two experiments in which calculators are used are described. In the first, the relative atomic mass of magnesium is determined. In the second, a constant for gaseous concentrations of two reactants and the product at equilibrium are determined

Holdsworth, David K. High Resolution Mass Spectra Analysis with a Programmable Calculator. Journal of Chemical Education 57: 99-100; February 1980.

Hughes, B. G. and Bundschuh, J. E. The Use of a Hand-held Programmable Calculator in Evaluating Freshman Experiments. Journal of Chemical Education 55: 336-337; May 1978.

The use of a programmable calculator for evaluating a student's performance on a quantitative laboratory experiment is reviewed.

McWilliam, I. G. Programmable Calculators. Journal of Chemical Education 51: 482-484; July 1974.

The use of programmable calculators for the simulation of experiments is discussed, and five examples of specific applications are given.

Runquist, Olaf et al. Programmable Calculators: Simulated Experiments. Journal of Chemical Education 49: 265-266; April 1972.

Simulated chemistry experiments with a Wang 360 programmable calculator are described, and data on a sample titration simulation are provided.

Seymour, M. D. and Fernando, Quintus. Effect of Ionic Strength on Equilibrium Constants. Journal of Chemical Education 54: 225-227; April 1977.

An experiment examining the effect of ionic strength on equilibrium constants is described. The experiment involves the use of a programmable calculator and the concepts of activity and activity coefficients.

Shearer, Edmund C. Applications of a Programmable Calculator in a Freshman Laboratory. Journal of College Science Teaching 5: 244-245; March 1976.

Use of a programmable calculator for student experiments, in grading laboratory reports, and in assigning accuracy and precision scores is discussed.

Snadden, R. B. and Runquist, O. Simulated Experiments. Education in Chemistry 12: 75, 77; May 1975.

A programmable calculator is used as a data-generating system for a simulated experiment involving conductimetric titration of an aqueous solution of HCl with an aqueous solution of NAOH.

College Demography

Sandery, P. The Programmable Calculator as an Aid. South Australian Science Teachers Journal 742: 49-51; July 1974.

A program is described which can be used to explore the effects of various values of R and C on population growth.

College Economics

Addis, G. H. The Use of Programmable Calculators in the Teaching of Economics, Part I. Economics 14: 3-8; Spring 1978.

Two calculator programs for computer-based economic simulations are described, each of which gives objectives, operating instructions, notes to the teacher, and detailed instructions for students.

Addis, G. H. The Use of Programmable Calculators in the Teaching of Economics, Part II. Economics 14: 50-58; Summer 1978.

The complete program for exploring the dynamics of the Harrod-Domar equation is given, and some statistical uses are mentioned.

College Geology

Shea, James H. Treatment of Earthquake Hypocenter Data with a Programmable Calculator. Journal of Geological Education 21: 29-34; January 1973.

Three investigations are developed using earthquake data and a calculator system with a card reader, X-Y plotter, and printout device. The investigations involve determining the spatial distribution of earthquake hypocenters with the goal of having students work with realistic data.

College Mathematics

Eisberg, Robert. Programmable Pocket Calculators in College Science Teaching. Journal of College Science Teaching 7: 305; May 1978.

The use of programmable calculators to solve second-order differential equations is discussed in an article which consists of examples selected by the author from his 1976 book.

Gazdar, Abdus Sattar. A Short Program for Simpson's or Gazdar's Rule--Integration on Handheld Programmable Calculators. Two-Year College Mathematics Journal 9: 182-185; June 1978.

A flowchart and program for numerical integration are presented.

Gerald, Curtis F. Interactive Computing with a Programmable Calculator: Student Experimentations in Numerical Methods. 1973. ERIC: ED 082 470. (8 pages)

In this paper presented at the June 1973 Conference on Computers in the Undergraduate Curricula in Claremont, California, the advantageous use of the Compucorp Model 025 programmable calculator in courses at California Polytechnic State University at San Luis Obispo is discussed. Students learned to solve non-linear equations and differential equations and were able to perform mathematical experiments analogous to the laboratory experiments of the physical sciences.

Hallden-Abberton, Patti and Waits, Bert K. The Programmable Calculator--An Inexpensive Teaching Machine. MATYC Journal 12: 215-219; Fall 1978.

A program for a programmable calculator is presented which permits evaluation of students' arithmetic computational skills.

Kruse, Harry Rudolph and Burkett, Hugh Alan. Investigation of Card Programmable and Chip Programmable Pocket Calculators and Calculator Systems for Use at Naval Postgraduate School and the Naval Establishment. Unpublished Master's thesis, Naval Postgraduate School, Monterey, California, March 1977.

The usefulness of card-programmable hand-held calculators in the management curricula of the Naval Postgraduate School and in the fleet were investigated. It was concluded that calculators provide significant advantages in teaching or learning mathematical concepts and that they are potentially important management and tactical support tools Navy-wide. In addition, the user's overall analytic capacity is improved.

Mohrman, Kathryn (Ed.). Innovations in Science Teaching. The Forum for Liberal Education, Volume II, Number 4, February 1980. ERIC: ED 181 844. (11 pages)

Curricular development in undergraduate programs in the biological, physical, and mathematical sciences at a number of colleges and universities is described. Included is The Ohio State University's program for teaching calculus with programmable calculators.

O'Loughlin, Thomas. Using Electronic Programmable Calculators (Mini-Computers) in Calculus Instruction. American Mathematical Monthly 83: 281-283; April 1976.

An experiment is described in which a minicomputer was used as an instructional aid in a calculus classroom and as a laboratory device.

Papers Presented at the Association for Educational Data Systems Annual Convention, Phoenix, Arizona. May 1976. ERIC: ED 125 658. (93 pages)

Included among papers on the use of computers and electronic equipment in instruction is one paper on the use of programmable calculators for calculus instruction.

Peckham, Herbert D. and Weir, Maurice D. Introduction to the TI-59 Programmable Calculator. Calculators/Computers Magazine 2: 52-57; September/October 1978.

Activities designed to familiarize calculus students with the use of the TI 59 are presented, and instructions for creating a library of programs on magnetic cards are given.

Schlaphoff, Carl W. CAI on a Programmable Calculator. MATYC Journal 9: 42-46; Winter 1975.

A procedure is described for presenting routine practice problems on a programmable calculator with attached teletype. The program uses a random-number generator to write problems, gives feedback, and assigns grades according to the procedures outlined and flow-charted by the author.

Sloyer, Clifford W., Jr. and Tingey, Henry B. Binomial Probabilities and the Pocket Calculator. Math Sciences Roundtable 1: 27-31; 1979.

Use of the calculator in computing probabilities for the binomial is discussed, with specific examples and problems.

Snover, Stephen L. and Spikell, Mark A. Because of Programmable Calculators, Why Avoid These Problems Any Longer? 1979. ERIC: ED 170 114. (11 pages)

Also appears as: Problems Now Solvable in Calculus and Other Beginning Undergraduate Mathematics Courses with the Use of Programmable Calculators. In Looking at Calculus: Perspectives for Teachers (edited by A. David Burdoin). Milton, MA: Association of Advanced Placement Mathematics Teachers, 1980. pp. 31-40.

Several examples are given of the types of nonstandard problems that students can solve by using programmable calculators. Finding limits, evaluating and calculating infinite series, computing variable length products, searching for data, and developing proofs are among the examples.

Wolfe, D. B. Natural Frequencies and Mode Shapes of Multi-Degrees of Freedom Systems on a Programmable Calculator. R.C.A. Reviews 39: 604; 1978.

Zimmerman, Mark. Random Numbers and Pocket Calculators. Calculators/Computers Magazine 2: 42-43; November/December 1978.

A procedure for generating random numbers with an HP 55 programmable calculator is presented.

College Physical Education

Miller, Doris I. Simulation of Sports Techniques by Digital Computer and Programmable Calculator. Journal of Health, Physical Education and Recreation 45: 65-67; March 1974.

Use of the programmable calculator in biomechanics research and in physical education classes is discussed.

College Physics

Albergotti, J. C. Instructional Uses of the Computer: Satellite Orbits on a Programmable Calculator. American Journal of Physics 41: 114-116; January 1973.

Maddock, M. N. and Power, Colin N. (Eds.). Research in Science Education, Volume 6. Proceedings of the Annual Conference of the Australian Science Education Research Association (7th, the University of Newcastle, New South Wales, May 17-19, 1976). 1976. ERIC: ED 143 501. (142 pages)

Among these papers is one on the effects of the programmable calculator on attitudes towards physics.

Phillips, R. F. Simple Gravitation Using a Programmable Pocket Calculator. Physics Education 12: 360-363; September 1977.

Calculation, using a programmable calculator, of the potential of the earth's gravitational field strength and the energies of satellites in orbit around the earth is described.

Pitcairn, Cameron C. and Baker, Gregory L. The Rocket Game. Physics Teacher 12: 427-429; October 1974.

A program for a programmable calculator is provided which simulates the problems of rocket propulsion, hovering, and soft landing.

Reiland, Robert J. A Realistic Model Rocket Program for a Small Programmable Calculator. Calculators/Computers Magazine 2: 72-74; September/October 1978.

A program for the prediction of the altitude achievable by a model rocket is given.

Schmidt, Stanley A. Fourier Analysis and Synthesis with a Pocket Calculator. American Journal of Physics 45: 79-82; January 1977.

Two programs for performing Fourier analysis and synthesis with a Hewlett-Packard (HP 25) calculator are described.

Summers, M. K. Programmable Calculators as an Aid in Physics Teaching. Physics Education 13: 246-250; May 1978.

Use of a programmable calculator to solve two kinds of differential equations (those defining simple harmonic and quantum harmonic motion) is described.

E. Games

Calculator Programs. Sky and Telescope 54: 292; October 1977; 55: 102; February 1978; 55: 301; April 1978.

Selected programs of astronomical interest that have been written for calculators are noted. Topic and source are indicated.

Dunlop, David L. Solitaire Mastermind with Programmable Hand Calculators. Calculators/Computers Magazine 2: 31-36; May 1978.

This game, programmed for an HP 25 calculator, can be modified for use with other programmable calculators. Using deductive logic, one or two players attempt to determine the code. Two programs, with variations, and a recording sheet are provided.

How to Program Calculators for Fun and Games. Popular Electronics 11: 39-46; June 1977.

A collection of six games for the programmable calculator are presented: Battle of the Dive Bomber, Football, Blackjack, Space Flight, Biorhythm Forecast, and Test Your ESP. Goals and rules are described, with programs for the HP 25.

Johnston, David W. Reactions (to the Tin Can Problem). Calculators/Computers Magazine 2: 45-47; February 1978.

In this letter, a programmable calculator program to solve the equation in the Tin Can Problem (Clyde, 1978) is presented.

Johnston, David W. Letters. Calculators/Computers Magazine 2: 82-83; April 1978.

Three programs for exercises in Scott (1978) are given in this letter.

Johnston, David W. Letters. Calculators/Computers Magazine 2: 12-13; September/October 1978.

A calculator program to solve for mean, standard deviation, and "normal curve" equation is given, to supplement Crothamel (1978).

Oglesby, Mac. "Hilo," "Hurkle." Calculators/Computers Magazine 1: 42-47; May 1977.

Directions for two calculator games are given. One can be played on either a four-function or a programmable calculator, the other only on a programmable calculator. Flowcharts and programming steps are provided.

Oglesby, Mac. Frogs. Calculators/Computers Magazine 1: 5-8; October 1977.

Rules to play the game "Frogs" on an SR 52 programmable calculator are given. Flowcharts, a program listing, and a sample game are included.

Wavrik, John J. Finding the Klingon in Your Calculator. Calculators/Computers Magazine 2: 29-33; January 1978.

A calculator program for finding a Klingon spaceship is given.

Wavrik, John J. Comments and Teacher's Notes and Answers. Calculators/Computers Magazine 2: 74-76; February 1978.

Answers for Wavrik's January 1978 article are given with comments.

Wavrik, John J. Shooting the Klingon in Your Calculator. Calculators/Computers Magazine 2: 62-64, 92; September/October 1978.

A program for the HP 25 calculator is given, to extend the activity presented by Wavrik in January 1978.

F. Other Uses

Bradshaw, M. Eugene. Programmable Calculator as a Test Controller. Machine Design 50: 64-65; April 20, 1978.

Garst, John F. Grade Analysis with a Programmable Pocket Electronic Calculator. Journal of Chemical Education 54: 114; February 1977.

Advantages of using a programmable calculator in computing student grades (e.g., in figuring weighted averages and in the linear adjustment of raw scores) are pointed out.

Peters, William F. The HP 25 as a Digital Clock and Timer. Popular Electronics 11: 57-58; August 1977.

How to program an HP 25 calculator to serve as a clock/timer with display in hours, minutes, and seconds is shown.

Reubens, Arthur. Trackin Down Equation Roots. Machine Design 50: 120; August 10, 1978.

A program for an HP 55 calculator is given.

Rowe, A. J. Machine-Marking of Multiple-Choice Tests: A Simple and Inexpensive System Using a Desk-top Calculator. Journal of Biological Education 6: 13-16; February 1972.

Schade, Herbert C. A Comparison of Student Characteristics Between Two Academic Years, 1971-72 and 1974-75. Institutional Research Report 3-75. 1975. ERIC: ED 130 722. (31 pages)

Statistical comparisons were made between 23 characteristics of students enrolled at Crowder College (Neosho, MO) during two academic years. A Chi-square program for the data was written for a Hewlett-Packard programmable calculator. The program is appended.

Tiny Computers Speed Business Decisions. College Store Journal 44(3, Sec. 1 of 2): 116-117; April/May 1977.

The use of programmable calculators in investment analysis, production scheduling, inventory control, and figuring compound interest rates is discussed.

Tweddale, R. Bruce. Difficult Budgetary Decisions: A Desk-top Calculator Model to Facilitate Executive Decisions. 1976. ERIC: ED 126 846. (16 pages)

This paper, presented at the Annual Forum of the Association for Institutional Research (Los Angeles, May 1976), describes a budgetary decision model developed to aid executives in arriving at tentative decisions on enrollment, tuition rates, increased compensation, and level of staffing. The model utilizes a Carnegie Model [illegible] programmable [illegible].

C. General

1977 Buying Guide to Electronic Calculators. MATYC Journal 12: 229-23?; Fall 1978.

[illegible] features, programming capability, and other factors for a wide variety of calculators.

Bl[illegible]ett, James. The Minimum Computer. Creative Computing 2: 22-24; November-December 1976.

The question, "At what level of complexity does a programmable calculator become a computer?" is discussed. Features of the Hewlett-Packard HP 25 are described.

Free, John. Specialized Calculators--Preprogrammed to Solve Programs Faster. Popular Science 209: 58-60; November 1976.

Advantages of preprogrammed calculators are discussed, and several models are described. Features of business or financial calculators are mentioned.

Free, John. Those Work-Saving, Problem-Solving Programmable Calculators. Popular Science 210: 64, 66, 70; February 1977.

Three groups of programmable calculators are identified: key programmable with volatile memory, card programmable, and key programmable with nonvolatile memory. Eleven calculators are compared on the following features: program steps, branching, addressable memories, logic, stack registers (reverse Polish notation), parenthesis levels, pending operations (algebraic operating system), and price.

Gottheimer, Debra. More Power to the Calculator. Administrative Management 39: 34-44; July 1978.

Programmable calculators, including hand-held as well as desk-top models, are described. Two charts are included, plus five industry trends.

Karp, Stewart. Calculators for the Chemist. Journal of Chemical Education 52A: 316-330; July 1975.

In the first of two articles, calculators of interest to the chemist are surveyed and their capabilities discussed.

[illegible] Journal of Chemical [illegible]

[illegible] described, including manufacturers, prices, [illegible]

[illegible] May 1976.

[illegible] 72-step programmable calculator [illegible] described, with a parts list and schematic diagram of the calculator included. Three examples (hyperbolic sine, [illegible]) [illegible] worked.

[illegible], Forrest M. Here Are the New Programmable Calculators! Popular [illegible] 29-35; May 1976.

[illegible] notation and algebraic methods of entry, and branching and conditional-comparison capabilities of programmable calculators are discussed. A shopper's guide to four [illegible] is provided. Sample programs for determining volume of a cylinder, for incrementing a number, for squaring [illegible] integers with a display of each result, and for a [illegible] game are included.

[illegible], Thomas Patrick. Significant Developments in the Use of Computers in School Mathematics: A Sourcebook for Administrators, Teachers, and Teacher Educators. 1974. ERIC: ED 108 910. (158 pages)

This doctoral dissertation (Columbia University) provides a critical review of developments in the use of computer extended instruction (CEI). Time-sharing, minicomputers, and programmable calculators are described and their usefulness in meeting educational needs [illegible].

Pederson, D. O. and Sharrah, Paul C. The Calculator Race--1979 Is Prime. Physics Teacher 17: 250-251; April 1979.

The TI 59 programmable calculator (with a PC 100A printer) and the [illegible]PET minicomputer are compared and their relative merits discussed.

Wavrik, John J. The Case for Programmable Calculators. Calculators/Computers Magazine 2: 63; April 1978.

Advantages of the calculator over the computer for certain uses are discussed.