DOCUMENT RESUME

ED 234 741 IR 010 807

AUTHOR	Edwards, Judith B.; And Others
TITLE	Computer Applications in Instruction: A Teacher's Guide to Selection and Use.
INSTITUTION	Northwest Regional Educational Lab., Portland, Oreg.; State Univ. of New York, Albany. Research Foundation.
SPONS AGENCY	National Inst. of Education (DHEW), Washington, D.C.; National Science Foundation, Washington, D.C.
REPORT NO	ISBN-0-89466-280-5
PUB DATE	78
CONTRACT	PES-7309325
NOTE	220p.; Foreword by Richard Otte.
PUB TYPE	Collected Works - General (020) -- Guides - Non-Classroom Use (055)
EDRS PRICE	MF01/PC09 Plus Postage.
DESCRIPTORS	*Computer Assisted Instruction; *Computer Programs; Elementary Secondary Education; Evaluation Criteria; Input Output Devices; *Media Selection; *Programed Instructional Materials; Programing; *Programing Languages
IDENTIFIERS	*Computer Uses in Education

ABSTRACT

Intended for upper elementary and secondary teachers in all subject areas, this guide provides practical advice on determining the appropriate application of computer technology and on the selection of specific, subject-related computer-based instruction units. Under the heading of hardware and software, topics discussed include communicating with a computer, computer size, getting and using user's programs, translators and programming languages, and the elements of a program. The computer is analyzed in terms of its uses as instructor, laboratory, calculator, object of instruction, and instructor's aide. The selection of computer-based instructional units is described from the beginning of the selection process through final decision making. Individual papers then focus on uses of the computer in the following specific areas: art education, business education, instruction for the deaf and hard of hearing, elementary school, language arts, mathematics curriculum and instruction, music education, physical education, secondary science, natural science, and the social sciences. "Keys to Recognizing General Purpose Languages," and some primary sources of computer-based instructional units, are appended. (LMM)

COMPUTER APPLICATIONS IN INSTRUCTION:

A Teacher's Guide to Selection and Use

Judith B. Edwards
Antoinette S. Ellis
Duane E. Richardson
Donald Holznagel
Daniel Klassen

Foreword by
Richard Otte
Project Officer
Technological Applications Division
National Institute of Education

Director, Computer Technology Program
Judith B. Edwards

Northwest Regional Educational Laboratory
710 S.W. Second Avenue
Portland, Oregon 97204

TimeShare
A Houghton Mifflin Company
Hanover, New Hampshire 03766

Some of the materials incorporated into this work were developed with the financial support of the National Science Foundation, Grant PES-7309325 to the Research Foundation of State University of New York. The grant was entitled Development of Computer-Simulation materials and was under the direction of Dr. Ludwig Braun, Professor of Electrical and System Engineering, State University of New York at Stony Brook.

ISBN: 0-89466-280-5

CONTENTS

FOREWORD

The role of the classroom teacher becomes increasingly demanding and complex as we understand more about how people learn and as tested technologies become available. Learning is a highly individual experience, an active rather than a passive act, with the learner seeking the information, the structure, and experiences, and the means he or she may need to understand and internalize new information. It is increasingly important that students be provided with a variety of instructional resources that incorporate multiple learning styles and modes of interaction. This multiplicity of resources increases the probability of a match between what is available and what the learner is seeking. The computer provides a variety of instructional strategies and delivery modes, and can be a useful and exciting means for expanding learning opportunities.

The cost of education increases daily. At the same time, the cost of computer technology has continued to drop dramatically. In fact, complete "home computers" offer programs ranging from arithmetic tutoring and biorhythm charts to simulated science lab experiments and computer based games such as Star Trek. These computers are broadly available for modest costs that are usually less than the cost of color TV sets. The implications for education are intriguing. In five years, the same computer capability will undoubtedly be notebook-size—or smaller like hand calculators are today.

The challenge to the classroom teacher who anticipates the utility of such technology is to become familiar with it and to learn to incorporate it effectively in his or her subject matter areas. How can teachers determine which of the many instructional units available are appropriate and useful for their students to experience?

This book deals with these concerns. It is intended for upper elementary and secondary teachers in all subject areas. It provides a practical guide to determining the appropriate application of computer technology and to the selection of specific subject-related computer based instruction units. The reader will become acquainted with the rudiments of computer hardware, computer programming languages, the different roles the computer can play in instruction, and how the computer can be used in several different curriculum areas for evaluating and selecting application units for instruction.

Development of this book by the Northwest Regional Educational Laboratory was sponsored by the National Institute of Education as a part of its continuing commitment to the sponsoring of research that results in the improvement of education through the application of modern technology.

Richard B. Otte
Education Technology Specialist
National Institute of Education

THE ESSENTIALS OF HARDWARE AND SOFTWARE

A COMPUTER SYSTEM: HARDWARE + SOFTWARE

TO THE USER

The two basic components of any computer system are the computer's physical equipment and the computer programs. The physical equipment is *hardware* and the programs are *software*.

Hardware

As illustrated in Figure 1–1, the main pieces of hardware in a computer system are:

INPUT/OUTPUT DEVICE
COMPUTER
AUXILIARY STORAGE DEVICE

The input/output device, or terminal (commonly a teletypewriter, as pictured in Figure 1–1) is used for communicating with the computer, both to input information and to receive information the computer prints out. The computer itself (called the Central Processing Unit, or CPU) is often located some distance from the input/output terminal. All CPUs include within their own internal circuitry some storage capability, called *working storage*, and most use additional storage located externally in separate auxiliary storage devices such as the disk drive in Figure 1–1. Further on in this section we will look more closely at different kinds of terminals and auxiliary storage devices and at CPUs of various sizes and capabilities.

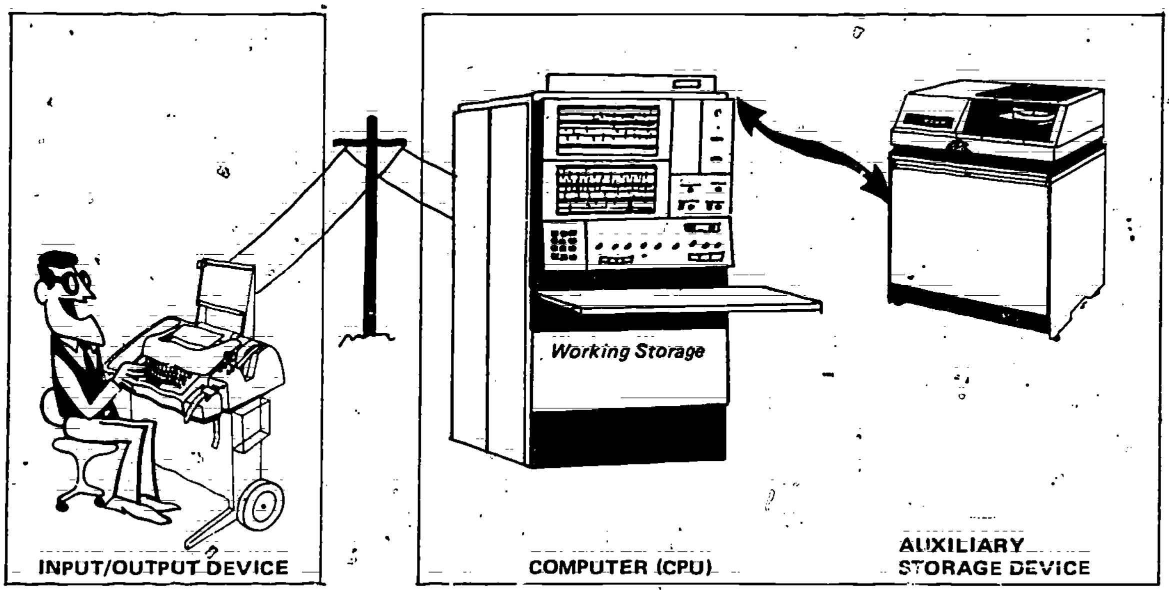


Figure 1-1. Computer hardware.

Two Types of Software

The software or programs of a computer system are sets of instructions that tell the computer what to do. There are two general types:

COMPUTER'S SOFTWARE SYSTEM (programs directing the computer's basic operations and general functions)
USER'S PROGRAMS (programs instructing the computer in specific applications)

Examples of the two types of software are listed in Figure 1-2.

COMPUTER'S SOFTWARE SYSTEM	USER'S PROGRAMS
ARITHMETIC OPERATIONS	PROBLEM-SOLVING
LOGICAL OPERATIONS	STATISTICAL ANALYSIS
PROGRAM SEQUENCING	TUTORIAL
ERROR CHECKING	DATA PROCESSING
TRANSLATING	SIMULATION
CONTROL PROCEDURES	CALCULATION
.	DRILL
.	GAMES
.	.
	.
	.

Figure 1-2. Two types of software.

The computer's software system is usually developed and written by the manufacturer. It is so basic to the computer's operation that it is included when you buy a computer, and as soon as the system is installed it is put into computer storage, where it remains permanently, to be called on and used continually by the computer.

The system may include translator programs which are designed to aid the user in expressing problem procedures in programs. They are not always necessary for the operation of the hardware, however, and are sometimes provided as options at extra cost.

The user's software, or programs, may be obtained from various sources: a user, users' groups, the manufacturer, or a software design firm. They may be put into storage at any time and kept there for as long as desired. When you begin working with a stored program, you simply call it up from storage. Programs are placed in storage under short-code names and kept there ready for use.

To get a quick picture of a complete computer system, imagine Figure 1-1 expanded so that the auxiliary storage device shows some of the software stored in it. (Remember that software may be stored either in working storage within the CPU or in one or more separate auxiliary storage devices, as shown in Figure 1-3.)

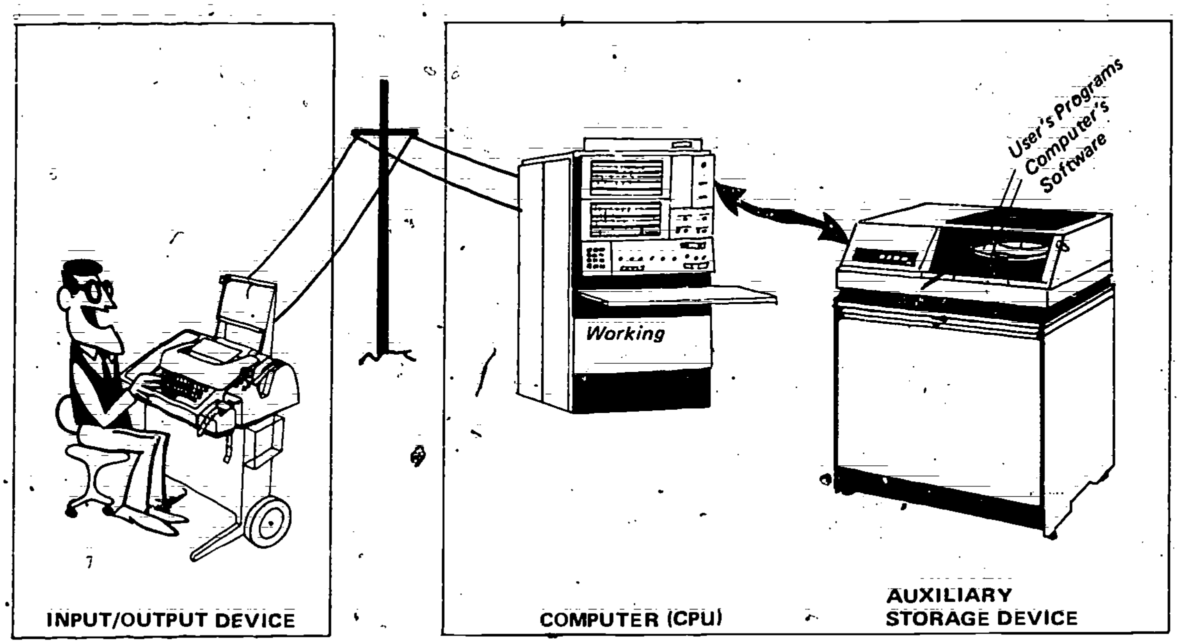


Figure 1-3. A Computer System: Input/output, CPU, and storage loaded with software.

In sections [specific reference] we discuss software in more detail, including the topics of getting and using programs and some elements of programs and programming languages. At this point, let's turn our attention to the question of how we use the hardware to communicate with the computer.

MODES OF COMMUNICATING WITH A COMPUTER

There are two ways in which the user can communicate with a computer—interactive and batch mode. These involve distinct processes and programs, and we will consider them one at a time.

Interactive Mode

If you have worked through one of the Sample Units in this course, you have already experienced interactive mode. The user has a dialogue with the computer which is like a telephone conversation—you communicate something, the computer responds, and the dialogue goes on back and forth—hence the term interactive. Figure 1–4 illustrates a typical interactive exchange.

From your experience with this mode, you can see the advantages of direct interaction with the computer. It probably allows the program to guide you in what data to enter and when; it allows for immediate correction if you enter inappropriate data; and it provides immediate results for you to consider and use in further exploration if you want to go on.

Interactive mode is very convenient and efficient for the user. It may, however, be expensive, since it always entails considerable time for the computer to print out instructions (questions and

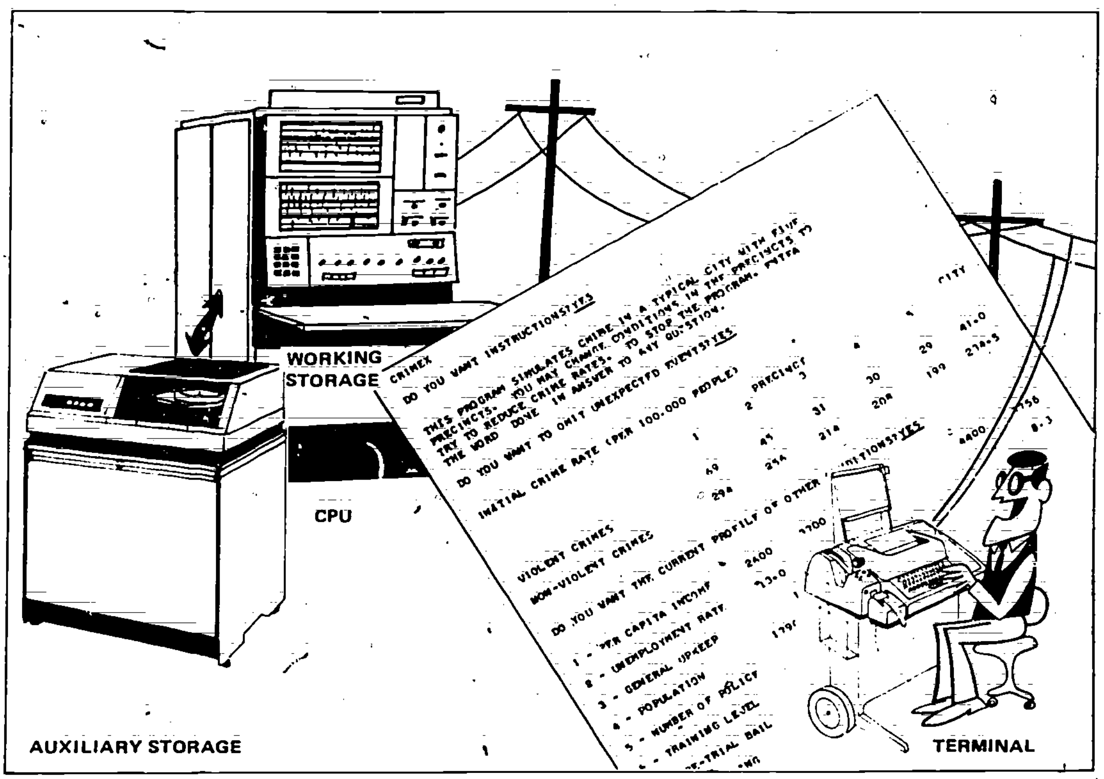


Figure 1–4. Using the computer in the interactive mode.

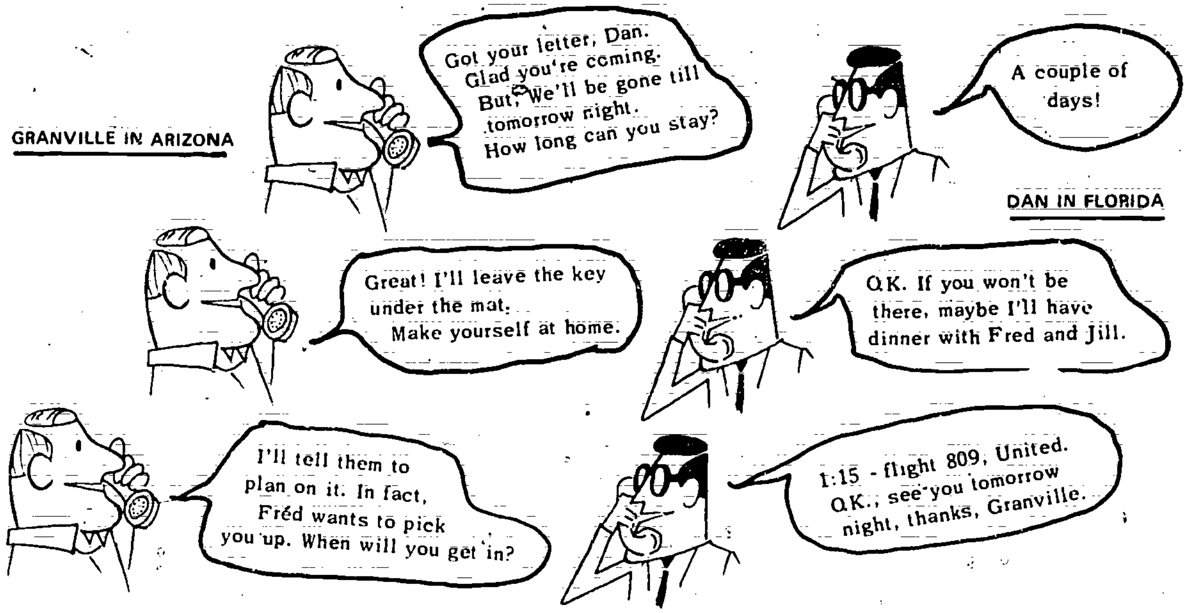


Figure 1–5. Example of an interactive conversation.

answers) and additional time while the user thinks of and inputs his responses. With cost of interactive use ranging from 1¢ to 10¢ or more per minute, cost can be an important consideration when evaluating an interactive unit.

The efficiency and expense of interactive mode find a ready parallel in long distance telephoning. There are less expensive ways to exchange information (e.g. by mail), but often the need for immediate information, or the need for back-and-forth exchange, makes it the most effective means of communicating. This is especially true when one response will depend on the other's previous response, as illustrated in the interactive conversation in Figure 1–5.

If Dan and Granville had relied on mail alone, it would have cost them less than long-distance phoning. However, it would have taken a week or more to exchange the information, and the day of Dan's visit to Arizona would have come and gone before Granville's first response could reach Dan's mailbox in Florida.

Whenever there is a need for immediate results, then, or for information to be exchanged with each response depending on a previous response, interactive communication may well be worth the cost. When time and interaction are not such pressing concerns, however, the more economical batch processing mode may be preferable.

Batch Processing Mode

Batch processing as a mode of communicating with the computer is more like letter-writing: it involves gathering together all your information and entering it or sending it off in one batch to be processed and responded to. All the results of the computer's processing are then returned to the user in one batch. An example of the sequence is shown in Figure 1–6.

In contrast to the immediate response characteristic of interactive mode, several hours or a

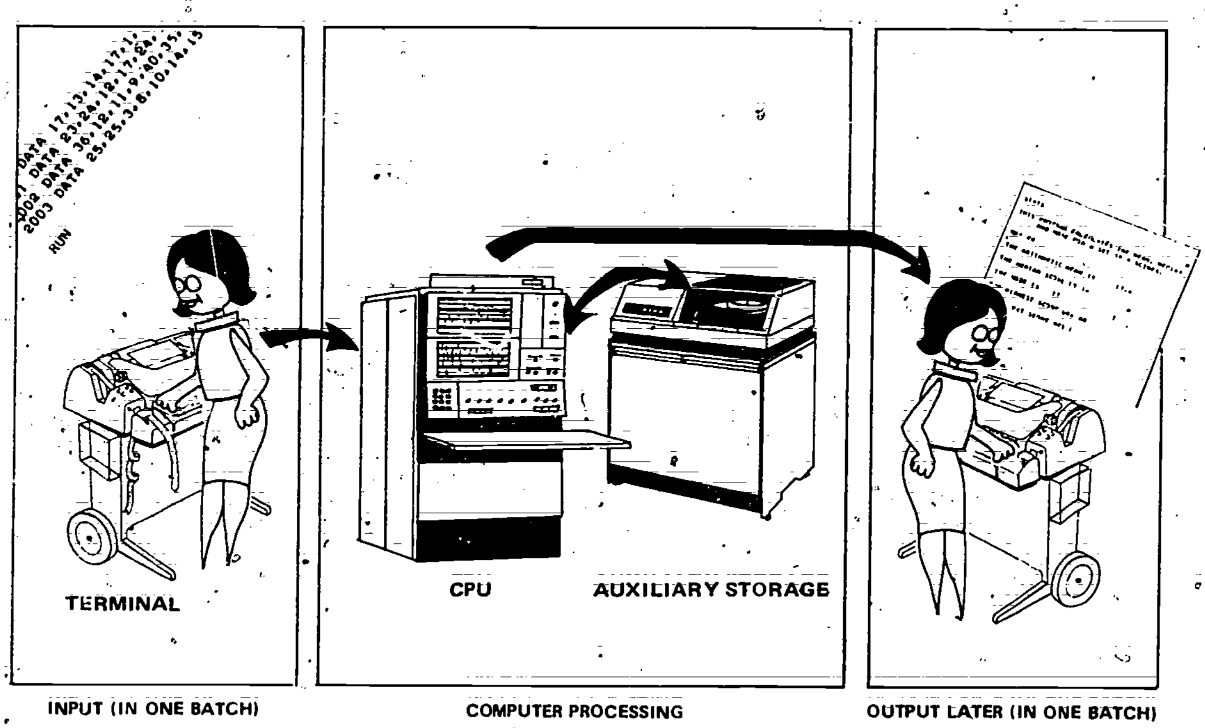


Figure 1–6. Sample batch processing sequence.

day may pass before a batched set of data gets to the computer for processing and the batched results are returned to the user.

Because all decisions about what is to be entered for processing have been made in advance, the actual computer time it takes to process the batch of data is often a small fraction of the time it takes to complete an interactive exchange with the computer. For this reason, batch is often more economical.

Comparing the Two Modes

Advantages and Disadvantages

The advantages and disadvantages of the two modes are summarized in Figure 1–7.

Program Characteristics

The characteristic difference between programs for the interactive and batch processing modes can be summarized as follows:

Interactive Throughout run: continuous interaction between user and program.
Batch Beginning of run: user enters all data in a batch.
End of run: user receives all results in a batch.

INTERACTIVE AND BATCH PROCESSING ADVANTAGES AND DISADVANTAGES	
INTERACTIVE	**BATCH**
Advantages • User actively involved. • Immediate detection of errors in entries. • Instantaneous feedback. Disadvantages • Relatively expensive. • Inefficient with large volumes of data. • More user time involved. • Programs usually longer than batch programs. • Data must be reentered every time the program is run.	Advantages • Large volumes of data handled efficiently. • Data can be saved and re-used. • Relatively inexpensive. • Many users can process data in a single batch. Disadvantages • User involvement is minimal. • Delayed information on errors. • May be time lag between entry and results.

Figure 1–7. Summary of advantages and disadvantages of the interactive mode and the batch processing mode.

Most computer-enhanced instructional materials are designed to be interactive. The active involvement of the user with the computer and the immediacy of feedback are strong motivational factors for students; in addition, the characteristic of leading a user from question to question (and answer to answer) makes interactive mode effective for many different instructional purposes, including problem-solving exercises, drill, and tutorial lessons. An important exception is in the area of business education and data processing instruction. Since most business and data processing uses of the computer are in the batch mode, instruction in this area is naturally batch-oriented. In any educational area, there are certain applications for which one mode or the other is more effective. In general, an interactive program is best suited to learning situations in which step-by-step progression and immediate feedback are desirable. Wherever very large amounts of data need to be processed and immediate results are not required (for example, in business education and statistics courses), batch processing is usually more efficient.

Programs written for the batch processing mode can usually be rewritten as interactive programs, and in many cases the reverse is also true. Therefore, when an instructor requires one mode of processing but wishes to use materials designed for the other, the program can usually be altered to suit the need. The teacher must judge whether instructional advantage is altered or lost in the exchange.

THE HARDWARE OF COMMUNICATION

Typically both interactive and batch modes require the use of an input/output device for communicating with the computer.

Input/Output Devices for Batch Mode

In the batch mode, either interactive or noninteractive input/output devices (terminals) are used to enter and receive data. The noninteractive terminals are usually referred to as "remote batch" terminals. Most of them include two pieces of hardware: a punched card reader for inputting data and a line printer for receiving the computer's output.

Most card readers accept either punched cards (IBM cards) or mark-sense cards* and will read and transfer electronically to the computer the data from the cards. A typical card reader (a Data 100 model) is pictured in Figure 1–8.

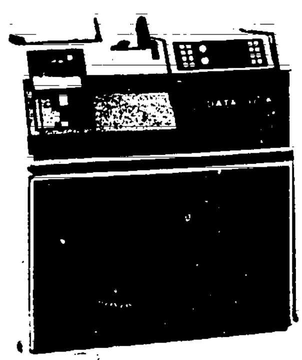


Figure 1–8. Typical card reader.

Similar to card readers are various kinds of page readers, designed to read input from much larger formats.

Line printers, such as the Centronics model pictured in Figure 1–9, receive and print out computer results very rapidly because they are designed to type an entire line at a time rather than one character at a time as typewriters do.

The most common interactive device, the teletypewriter, is usually equipped with a paper tape punch reader device, and the combination of typewriter capability and tape-reading capability make it usable for batch mode. Batched input can be punched onto paper tape and transmitted to the computer through the paper tape reader in one batch at the rather quick reading speed of the reader device, as is illustrated in Figure 1–15 on page 16. Where the paper tape punch/reader is not available, however, the teletypewriter itself can be used to input batched data

*Mark-sense cards are those which are marked by hand using a pencil.

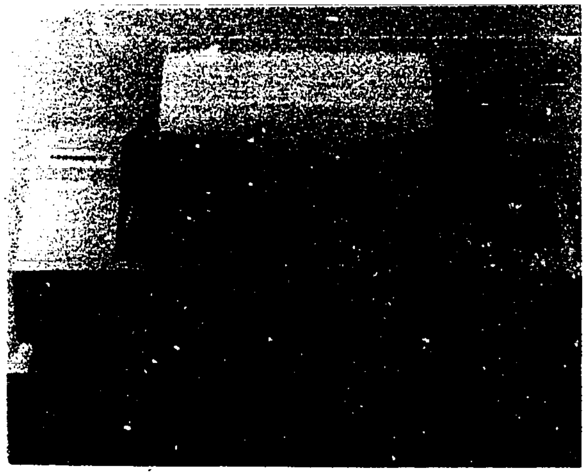


Figure 1-9. Typical line printer.

by the slower process of typing the data on the teletypewriter keyboard when the terminal is connected to the computer. Using this terminal, output can be received in a batch from the computer and typed out by the terminal's typewriter device.

Input/Output Devices for Interactive Mode

Several types of terminals are available for use in interactive mode. The most common are the various kinds of electric typewriter terminals (including the popular Teletype) and the video display terminal.

You are probably already familiar with the special electric teletypewriter called a Teletype; it is very much like the terminals used in telegraph offices and newsrooms. Its main features for

Figure 1-10. A Teletype.

interactive communication are (1) its typewriter keyboard and (2) its capability of printing out the messages you type as they are transmitted to the computer while typing out the computer's messages to you. Other kinds of typewriter terminals can also print input and output and transmit information to and from the computer. Some have additional features, such as upper and lower case letters and extra fast printing capability. Figure 1-11 shows one of these.

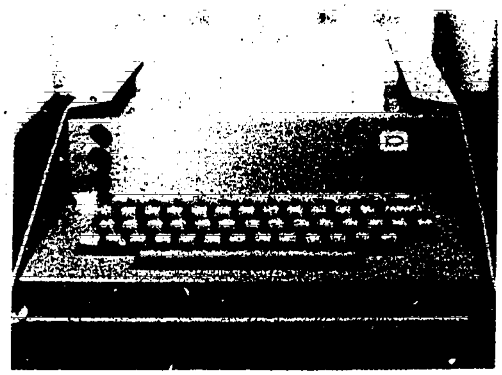


Figure 1-11. A typical thermal printer terminal.

This terminal uses heat to print on special heat-sensitive paper and can print very rapidly and quietly. The Computer Devices model in Figure 1-11 also has the feature of being small and lightweight, so that it can conveniently be transported.

The video display terminal, frequently called a CRT (cathode ray tube), has a keyboard and a TV-type display screen rather than typing paper. The letters and words you type are displayed on the screen as they are transmitted to the computer, and return messages from the computer (words, graphs, and so forth) are displayed on the screen. Video display screens may also be

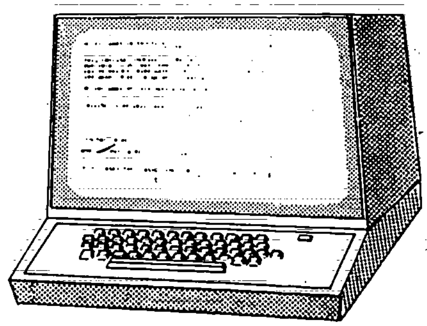


Figure 1-12. A video display screen.

equipped to accept input from special devices called light pens, which look like regular pens but have a small light bulb at the tip. The user may input information by pointing the light pen at appropriate data on the display screen, such as a selected answer from a list of choices or a section of a graph that is to be changed. One advantage of the CRT in the classroom situation is its quiet operation. Another advantage is that the computer's output to you can usually be displayed much more quickly on the CRT screen than it could be typed out by the electric teletypewriter. This is particularly true when the output is in the form of a graph. One disadvantage, on the other hand, is that once the output is erased from the display screen, it cannot be retrieved on most CRTs.

Connecting Terminals to Computers

Batch terminals, like card readers and line printers, are most commonly connected by permanent wiring directly to the central computer and need only be turned on to be connected to the CPU for input or output operations; interactive terminals, on the other hand, are usually connected to the computer by telephone lines. The computer is dialed on the telephone, and when the "answering" tone comes, the receiver is placed in the special coupler device hooked up to the terminal. Once the phone connection is made, the user must identify himself by typing in the correct code. The communication is then established, and the input and output travel between the terminal and the computer through the telephone lines.

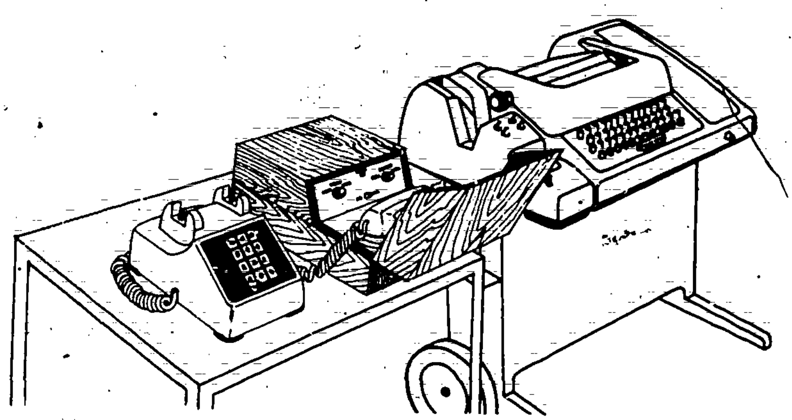


Figure 1-13. A teletypewriter connected to a telephone coupler.

A single computer can often have many terminals connected to it by telephone. When two or more terminals are connected to the computer simultaneously, it is called *time-sharing*. When you are sharing the time of a computer with other terminals, you usually can't tell. The users' programs are processed one at a time at very high speed, and the computer can normally handle inputs and outputs in such rapid succession that no delay is experienced for any user, and the user is not aware that he is sharing the time. Sometimes there may be a few seconds delay in the computer's response to your terminal, which may indicate that it was busy for those few seconds attending to another terminal.

THE QUESTION OF COMPUTER SIZE

Detailed aspects of computer size and type are frequent points of discussion, primarily among experienced users. In selecting and specifying instructional applications, however, a user need consider only a few of these when considering the relative advantages of minicomputers, microcomputers, and large computers.

Types of Computers Today

Minicomputers

When first developed, minicomputers were not only smaller physically than the conventionally used computers, but had much smaller computing capabilities. At the present point of development, however, it is sometimes difficult to determine the difference between minicomputers and conventional models because "minis" are performing some tasks as well as the larger machines have done in the past. It is common today to identify minis by the amount of information which can be contained in each basic storage element (usually the equivalent of two alphabetic characters) and to some extent by the expandability of working storage.

Microcomputers

A more recent development, the microcomputer (or microprocessor), is the result of new advances in miniaturizing electronic circuitry. An entire CPU may now be put on a chip that sells for $26 and is the size of your fingernail. Since microcomputers can be sold very cheaply, kits of components can be purchased for a few hundred dollars; a school can buy one for students to assemble themselves. Microcomputers now lack the sophisticated and proven software which has been developed and refined over the years for minicomputers, but that is changing rapidly. A microcomputer with sufficient storage, expansion capability, and software could perform as well as some minis.

Larger Computers

Before mini- and microcomputers entered the market, computer size was not a point of much discussion among users. Although computers varied in physical size and also in storage capacity, their accessibility and the functions they could perform were fairly standard to all users. For the most part, computers were (1) affordable only by organizations with a great deal of computer work to do and (2) locked up in rooms entered only by computer professionals. Users had no direct contact with the large machine. Now, of course, minis and micros are often affordable by small organizations, schools, and even individuals. In addition, users may gain access to the mini- or microcomputer through a terminal or even directly.

Meanwhile, the larger machines are still there and are still doing most of the work that is done by computers in the world. A large computer system can do a great deal more computer processing than a micro- or minicomputer and can typically serve hundreds of more input/output terminals simultaneously than the smaller computers.

Because of their greater cost it is not likely that small organizations, schools, or individuals will acquire a larger computer. A sizable school district, though, may need enough computer processing to warrant buying a large central computer.

Time-sharing Systems

A time-sharing system is a combination of computer hardware and the proper control software to allow more than one terminal to be served at the same time by the CPU. Because of the extremely high speed of operation, the computer's response to a time-sharing user usually makes him appear to be the only user. A time-sharing system may use a larger, micro- or minicomputer, and it often is composed of more than one computer (CPU). Hence, time-sharing systems vary in size according to the type and number of CPUs involved. A typical system based on a single minicomputer is commonly designed to handle up to 32 terminals simultaneously. Systems based on larger CPUs or those using several CPUs, have been known to handle over 300 terminals simultaneously.

Three Important Factors of Size

The point to remember in selecting and specifying instructional applications is not to make any assumptions about whether the computer can handle a given application simply by virtue of such a label as "mini." The main question is, is the computer big enough to run the programs you wish? Some of the relevant factors are: (1) interaction capability; (2) number of terminals; and (3) size of working and auxiliary storage.

Essentially, software is what determines whether a system can support an interactive program. A minicomputer with only one terminal can still have interactive capability. The number of terminals needed to make a particular application effective is sometimes a question. For example, if you wish to provide every student with arithmetic drill daily at a terminal, you will need many terminals, and therefore a time-sharing system will be required to support them.

The size of both working storage (inside the CPU) and auxiliary storage can be important. Every program requires some working storage each time it is run; the amount needed depends on the length of the program. The existence of enough auxiliary storage to contain a large program does not guarantee that working storage will be large enough to contain the program when you wish to run it, but there are ways of dividing programs to get around this. Auxiliary storage is a must for applications which are file-oriented, that is which need to store and/or replace large amounts of data from one usage of a program to the next. It is also required if it is important to retain one or more programs in storage rather than reload them each time. Consequently, the amount of auxiliary storage space available determines how many programs and files can be kept. Among the most common auxiliary storage devices are the magnetic tape drive, which reads data from magnetic tape, and the disk drive, which reads data from disks resembling conventional phonograph records. In both cases, the data are recorded onto the medium in the form of magnetic spots and are read by "read-heads" on the drive.

As you can see, then, computer size is sometimes irrelevant and sometimes crucial. The question of whether the computer to which you have access is big enough for your needs can best be answered individually for each program by someone familiar with your computer facilities. This person may be on the staff at the county school office or local district office, or wherever the computer you use is located; in your own school, it may be another teacher, often a math teacher. When you find computer-based instructional materials that are appropriate to your needs, the expert will be able to tell you fairly quickly whether your computer is able to run the program.

GETTING AND USING USER'S PROGRAMS

As you have seen, there are two types of software: the user's programs, which dir[illegible] the computer in special applications, and the computer's own software system (usually su[illegible]d by the manufacturer), which controls its basic operation. The sample programs included he[illegible] are user programs.

User's programs are of many different sorts because there are so many fields in which computers are used for such a wide variety of purposes. People in business use computers to process paychecks and billing statements; teachers use them to enrich their curricula; scientists make major use of them for calculating and predicting; libraries use them as cataloguing and retrieval tools; and so forth. Special programs are needed for each application.

Sources of User's Programs

A user may write his own programs (if he knows how); he may have an expert programmer write them or he may find suitable programs that already exist. Computer manufacturers carry extensive selections of user-oriented programs, and many major universities have computer projects with collections of user materials.

As a teacher, you may obtain programs in any of these three ways. As a start, you will probably want to make use of the growing number of programs already available from other sources. Available materials range in completeness from the listing of a single program to a total curriculum package, including such support material as a Teacher's Guide, Student Workbook, listing of the program, and sample runs.

In Part III, Selecting Computer-Based Instructional Units, you will learn how to evaluate and select the most useful programs and materials for your particular classes.

As a basis, you will need to know some fundamental things about user programs and how you can get them into your computer.

Loading Programs into the Computer

When looking at computer-based application units, you will often see little more than a program listing—a copy of the step-by-step instructions comprising the program itself. A short listing is shown in Figure 1-14. This is, of course, the basis of any curricular application using the computer. With the program listing in hand, you can load the program into your computer and run (use) it, either to evaluate it or to actually use it.

```
10 REM  THIS PROGRAM READS STUDENTS'
20 REM  NAMES AND GPA'S AND PRINTS
30 REM  AN HONOR LIST (GPA ABOVE 3.5)
40 DIM S$(50)
50 READ S$,G
60 IF S$="END" THEN 9999
70 IF G<3.5 THEN 90
80 PRINT S$,G
90 GO TO 50
9999 END
```

Figure 1-14. Example of a short program listing.

How is a program loaded into the computer? Three common ways are: typing at a terminal which is connected to the computer; loading a punched paper tape; or loading punched or marked cards.

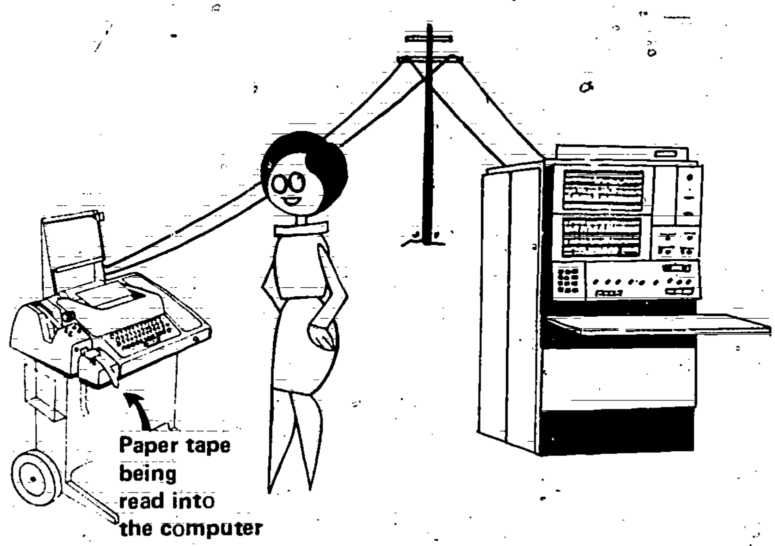


Figure 1-15. Loading a program from a paper tape into the computer.

Loading a program by directly typing it in at the terminal can be costly, since you tie up both a telephone line and the computer for the entire time you are typing. A less expensive way is to read in a punched paper tape, using the tape reader on your Teletype, as illustrated in Figure 1-15 or to read in a deck of punched (or marked) cards, using a card reader. To do this, you must first acquire the punched tape or cards, or arrange to have the tape or cards punched from the program listing. Sometimes the organization that supplied the program listing also supplies the necessary tape or cards, which saves you a step.

Once loaded into the computer, the program can be stored* and called out whenever it is needed. If storage space is limited, you need not store the program; you can save the paper tape or cards and reload the program each time you use it.

TRANSLATORS AND PROGRAMMING LANGUAGES

Let's look again at the sample program listing shown in Figure 1-14, reproduced in Figure 1-16. This example is a simple program written in a programming language called BASIC. Even if you don't know BASIC, you can easily interpret and understand words like READ and PRINT in the program. But computers cannot read and interpret these words; they can only work with machine language. Programs written in such English-like languages as BASIC, then, must be translated into the machine language that the particular computer is wired to understand. For example, the easily understood instruction "PRINT S$, G" would have to be translated into machine language before the computer could accept it. The translation might look something like this:

```
17 1028 001
05 6666
12 6666
```

*See pages 29-30, "Where Is the Program?"

```
10 REM  THIS PROGRAM READS STUDENTS'
20 REM  NAMES AND GPA'S AND PRINTS
30 REM  AN HONOR LIST (GPA ABOVE 3.5)
40 DIM S$(50)
50 READ S$,G
60 IF S$="END" THEN 9999
70 IF G<3.5 THEN 90
80 PRINT S$,G
90 GO TO 50
9999 END
```

Figure 1-16. Example of program listing.

The Translator Programs

A major part of the computer's own software system is its collection of translators, the programs that translate English-like programming languages (which most users use) into machine language (which computers use). Humans can, and do, learn to write programs in machine language, but it is a very tedious process, and it is far easier to write programs in English-like languages and let the computer's translator do the translating. As there are many different programming languages, a computer may have numerous translators in its software system—one for each user language the system can accommodate.

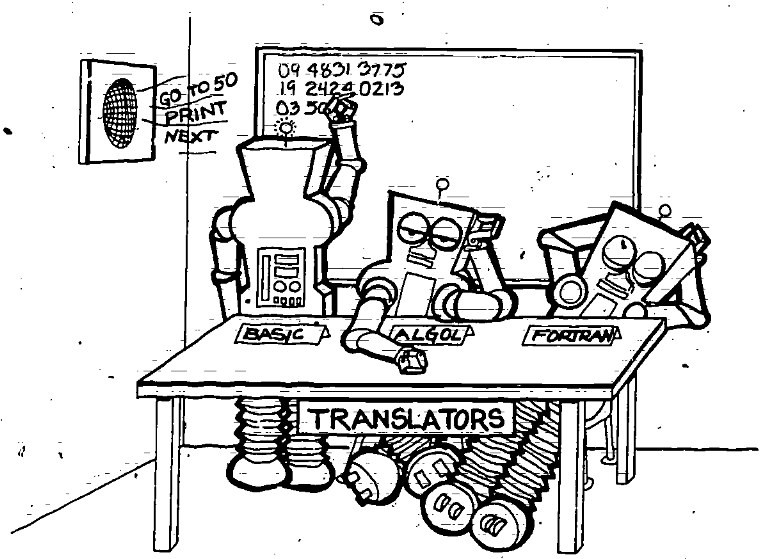


Figure 1-17. Translators: Part of the computer's own software.

Users must know whether their computer has a translator for the particular language they want to use; if not, they cannot load and run a program. For example, if you run across a program written in FORTRAN which you would like to use in your classes, you would first have to find out if your computer has a FORTRAN translator. Even if it has not, you still may be able to use the program: it could be translated by an expert into a language for which your computer does have a translator. Many FORTRAN programs, for example, can be rewritten in BASIC without much trouble; and conversely, most BASIC programs can be rewritten in FORTRAN.

General-Purpose Programming Languages

Each of the major fields of computer use—business, science, education—has its own particular vocabulary and computational needs. For creating computer programs to satisfy these needs, there exist today numerous general-purpose programming languages, each with specific characteristics that make it particularly useful in particular fields. Alongside these languages, some highly specialized languages have been developed for very specific areas of application. The following pages describe some of the major general-purpose languages in use today and a few of the special purpose languages, called author languages, developed for educational uses.

The common general-purpose languages used in education are BASIC, COBOL, and FORTRAN, though several others are also encountered. The following paragraphs give a brief description of the general characteristics of each. It is important to note that while some of these languages have been relatively standardized (that is, there is some agreement on vocabulary and characteristics), there are differences in details from computer to computer, so that a program written in standard BASIC (say) for one computer may require some changes to translate it for use on another computer. The differences are most often found in such special features as mathematics functions and file-handling commands. Another factor to keep in mind is that each general-purpose language may have several versions, each incorporating special features, requiring different amounts of computer storage.

BASIC

The programs you will encounter in computer-based curriculum units are most likely to be written in a language called BASIC (*B*eginners *A*ll-purpose *S*ymbolic *I*nstruction *C*ode). Developed at Dartmouth, BASIC has come to be the most universally used language for computer-based instruction. It is easier to teach, learn, and use than other languages, yet retains much of the power of the more mathematically oriented language, FORTRAN, and provides for easy handling of input and output. BASIC was designed for an interactive system and is often called a conversational language, because the programmer can provide for user interaction, while the program is running, in the form of answering questions or providing data. This characteristic gives the instructional designer the chance to implement an instructional dialogue. In general, BASIC allows for easy formating of output, an important characteristic when teaching beginners to program.

COBOL

COBOL (*CO*mmon *B*usiness *O*riented *L*anguage), has been developed to meet the needs of businesses. In the usual applications, computers are called upon to maintain and process files of information, produce reports, compute payroll, keep track of accounts receivable and payable, monitor inventory, and so on. Computation is usually trivial—the simplest arithmetic is generally all that is needed. COBOL is well suited to this environment. It is designed to facilitate the formating of reports, lists, and data summaries and the sorting and merging of large files of alphabetical names, etc. as well as numerical data. It is a batch-oriented language, and a COBOL program expects data from punched cards, magnetic tape, or disk files.

FORTRAN

The world's most widely used programming language, FORTRAN, was designed primarily as a scientific language. It is especially suited to mathematical and scientific endeavors, as the name FORTRAN (*FOR*mula *TRAN*slation) suggests. It has such special characteristics as trigonometric functions to make programming and processing involved calculation sequences faster and more efficient. It requires from the programmer more detailed definition of desired output format than BASIC, and therefore it is more difficult to learn. Although frequently used on-line from a terminal, it is not interactive (a program cannot pause for user responses to questions) and is therefore not suited to dialogue.

ALGOL

Another scientific language, more frequently used in Europe than in this country, is ALGOL (*ALGO*rithmic *L*anguage). Its function is similar to that of FORTRAN and its purpose is better suited to scientific calculations than to business resports.

PL/1

A fairly recent development is the language PL/1, first introduced by IBM as an attempt to combine the scientific features of FORTRAN and ALGOL with the fluency of COBOL. An additional feature is its versatility—it can be used with equal ease for interactive or batch processing. PL/1 gains greater acceptance in the computer profession year by year, although it is still featured primarily on IBM computers.

APL

*A P*rogramming *L*anguage, APL, was designed to be flexible and to allow concise expression of procedures in either a highly abstract, symbolic manner, for those who think in that manner, or more verbose expression (as in BASIC) for those who prefer to work with words. Thus it is a very nice tool for mathematics and science. But it requires a special terminal keyboard providing the necessary symbols. A single APL line can represent many lines of BASIC or FORTRAN. It is an interactive language, not suited to batch operation. Although designed at IBM, it is being used on other computers.

LOGO

The LOGO language is designed especially to facilitate the teaching of mathematical concepts. It allows for expressing computation and procedures in words, or strings of alphabetic characters, thus appearing particularly English-like. It is not as likely to be chosen by a teacher to program an application for student use, because the purpose of LOGO is to teach students to write programs themselves. It lends itself as a medium for dialogue, and is interactive; it is not useful for batch processing. As yet, LOGO cannot be used on as many makes of computer as the other general-purpose languages discussed above.

For a more detailed look at several of these general-purpose languages see Appendix I, Keys to Recognizing General-Purpose Languages.

Author Languages

Author languages are an especially useful group for teachers. In contrast to general-purpose languages, which are designed for straightforward input-process-output operations, author languages are designed to guide the author in entering lessons of all kinds and to handle the large volume of verbal presentations and responses typical of instructional interactions. Before author languages were developed, a teacher using BASIC or FORTRAN might have had to use five or six lines of complex strings of coded qualifiers and formating directives to enter a simple question in a lesson. Author languages typically provide the author with brief codes for presenting instructional events and allow him to enter the desired verbal material directly without complicated and time-consuming coding efforts. Some of these languages prompt and guide the author in entering his sequence of instructional activities, including instructional statements, questions to be presented, expected answers, appropriate responses, branches and so forth, according to the set format of the language.

An author cannot use an author language without prior study of the language and its requirements. Like other languages, author languages have syntax and a vocabulary (often symbolic) which require time and experience to master. A trained programmer may be needed to interpret the author/teacher's written lesson in the author language to be used.

Here are brief descriptions of some of the more commonly used author languages available today.

Coursewriter

One of the earliest examples of an author language, Coursewriter, originated on IBM equipment but has been transferred to some other brands. Originally, this language was used to computerize programmed instruction materials; now, however, it is capable of incorporating film, video, and audio instruction as well as printed material. One of the built-in features of this author language is that lessons structured in Coursewriter automatically record student performance data for teacher reference.

PLANIT

Since PLANIT incorporates a mathematical problem-solving language called CALC, this author language is most appropriate for authoring tutorial lessons involving problem-solving. Like Coursewriter, it is designed to keep records automatically of student performance for teacher reference. A notable characteristic of PLANIT is that it has been designed to be transferable to most computers, regardless of manufacturer, with the creation of a small special software package to interface between the PLANIT language and the computer. PLANIT can be used, however, only with larger computers today, because of special storage requirements; a few minicomputers can handle the PLANIT language but as yet no microcomputers have the necessary capacity.

PILOT

The PILOT language is an interactive author language which is easier to learn and use than either Coursewriter or PLANIT. As such, it has been used by elementary students to write story-generating programs. PILOT is also a versatile author language insofar as it has been put in operation on a wide range of computers, particularly minicomputers.

IDF

Interactive Dialogue Facility (IDF) was designed by the Hewlett-Packard Company specifically for use on its minicomputer-based timesharing systems. As is the case with the other two computer-specific author languages described below, DECAL and PLATO, IDF is made especially easy to use because entering an instructional dialogue into the system is more a matter of responding to prompts from the computer than writing a program. The authoring facility stores the text provided and supplies an automatic framework for presenting the lesson to a student and for recording student progress.

DECAL

The Digital Equipment Corporation author language DECAL operates on minicomputer-based time-sharing systems produced by that corporation. It is similar to IDF in that it provides the author with prompts from the computer when lessons are being entered.

PLATO

Developed at the University of Illinois under a large-scale grant from the National Science Foundation (NSF), PLATO is composed of a complete time-sharing system including a large computer, authoring software, and terminals. The PLATO terminal uses a special technology for the screen, enabling computer output to be displayed and photographic slides to be projected on it simultaneously from the rear. The terminal is used to prepare lessons as well as to present them to the student. Another feature is the capability of recording a vocabulary in voice for constructing audio responses and directions in lessons. Control Data Corporation now manufactures and markets the complete PLATO system.

TICCIT

Like PLATO, the TICCIT system was developed under an NSF grant. It is composed of a time-sharing system based on minicomputers. The terminal is quite different from that of PLATO, making use of a color television set as the output medium. It has a special keyboard with the command structure built in. Rather than write computer programs, an author uses the special keys to indicate specific actions. Students use the same keyboard to interact with a lesson and to control their own paths through the material. For example, it is the student's decision how many repetitions of a practice session are sufficient. TICCIT was developed jointly by MITRE Corporation and Brigham Young University and is now marketed by the Hazeltine Company.

Special Software for Special Devices

Many different electronic devices can be connected to computers and operated by electronic signals, and it is desirable to allow a user to control them through a program. Some devices particularly useful in instructional settings are plotters and music generators. A plotter is a small platform with a pen attached to an arm which is moved over the platform according to programmed instructions. It can be valuable in mathematics, science, and art. A related device is the Turtle (produced by General Turtle, Inc.), a small dome-shaped device on wheels with a pen positioned vertically in the center. It can be moved over a floor, table, or other surface in the same manner as the plotter pen on its platform. Another special device is a video terminal with graphic capabilities. This enables a program to control the motion of a dot on the screen with retention of the track of its motion, resulting in a picture.

All such devices require such special terms in the programming language as LEFT, RIGHT, PEN DOWN, PEN UP for a programmer to control them. If you wish to design or acquire such applications, you need to determine whether the language you have available contains the commands necessary to control the device you want. A language designed with specific allowance for these functions is LOGO. Some manufacturers provide a BASIC with built-in plotter functions.

THE ELEMENTS OF A PROGRAM

While you may never wish to become an expert computer programmer, you may be called upon to make a general evaluation of an instructional unit which is based on a computer program, or to interact with a programmer in developing your own computer-enhanced units. It will therefore be useful to know how to get a rough idea of what's happening in the program.

Input, Process, Output

First, let's review the function of a computer program. A computer program is the means by which a user communicates needs to the computer in a systematic, step-by-step way. In order to solve any problem, the computer essentially does only three things: it receives data as *input*, *processes* the information, and *outputs* new information. Any computer program, written in any

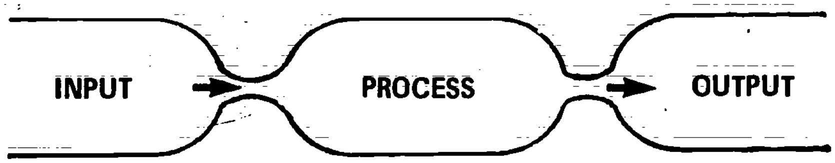


Figure 1-18. Input-process-output flow.

language, can and must reduce the problem it deals with to these three fundamental elements. Once we identify what is to be input, how it is to be processed, and what is to be output, we can see the purpose of a program quite easily.

Let's examine the very short BASIC program in Figure 1-19 below to identify each of the three elements.

```
30 INPUT X,Y
40 LET S=X+Y
50 PRINT S
60 END
```

Figure 1-19. Simple BASIC program.

Input Instruction

This program directs the computer to add two numbers and print the sum. The *input* instruction is easy to find in a BASIC program, since the word INPUT is used. The statement

30 INPUT X, Y

provides for the user to input (enter) data in the form of values for any two numbers, which the computer will call X and Y.

Process Instruction

The *process* instruction in this program is simply:

40 LET $S = X + Y$

which tells the computer to add the values for X and Y and call the sum S. Processing information usually involves some calculation or some rearranging of data, often (as in this case) in the form of one or more equations.

Output Instruction

The *output* instruction:

50 PRINT S

tells the computer to print out (on the teletypewriter) the sum S which it has just calculated.

End Instruction

Finally, the computer is told that it is finished and can stop:

60 END

Line Numbers

The numbers 30, 40, 50, and 60 are line numbers and identify or name each instruction in the BASIC program.

Comments

Programmers frequently insert remarks or comments in a program to make its purpose clear to anyone who may be examining it. In BASIC, such comments are identified by the word REM (for *rem*ark). Read over the same simple BASIC program, this time with "remarks" added:

```
10 REM  THIS IS AN EXAMPLE: A
20 REM  PROGRAM TO ADD 2 NUMBERS
30 INPUT X,Y
40 LET S=X+Y
50 PRINT S
60 END
```

Figure 1–20. Simple BASIC program with remark statements.

With the remarks added to the program, you can examine and determine its purpose. Many programs have such helpful explanatory remarks, though the word used to identify them will vary from language to language.

A Program at Work

Let's look now at how this program works in actual practice.

First the program must be loaded into the computer. Let's assume you are using an electric typewriter terminal and are loading the program from punched paper tape using the electric typewriter's paper tape reader. As the program is loaded, it will automatically be listed on the electric typewriter paper and will look like the listing below.

```
10 REM   THIS IS AN EXAMPLE: A
20 REM   PROGRAM TO ADD 2 NUMBERS
30 INPUT X,Y
40 LET S=X+Y
50 PRINT S
60 END
```

Once the program is loaded, you can direct the computer to execute it. On some computers this is done by typing the word RUN after the program is loaded.

```
10 REM   THIS IS AN EXAMPLE: A
20 REM   PROGRAM TO ADD 2 NUMBERS
30 INPUT X,Y
40 LET S=X+Y
50 PRINT S
60 END

RUN
```

When the computer executes the program, it ignores the REM statement, then encounters the INPUT instruction and types a question mark (?) to ask you to supply the two numbers as input data. You then enter the two numbers from the terminal by typing them on the keyboard. (The two numbers entered are underlined in the sample below.*)

```
10 REM   THIS IS AN EXAMPLE: A
20 REM   PROGRAM TO ADD 2 NUMBERS
30 INPUT X,Y
40 LET S=X+Y
50 PRINT S
60 END

RUN
?389, 877
```

Next, the computer processes the data (adds the two numbers) according to the instructions and prints out the new information—the sum.

*Throughout this text, user entries on samples of runs are underlined so that you can readily see them. No underlining will appear, of course, when you actually interact with programs.

```
10 REM   THIS IS AN EXAMPLE: A
20 REM   PROGRAM TO ADD 2 NUMBERS
30 INPUT X,Y
40 LET S=X+Y
50 PRINT S
60 END

RUN
?389, 877

1266
```

Then it prints out the word DONE. The complete printout would look like this:

Program loaded from paper tape	10 REM THIS IS AN EXAMPLE: A 20 REM PROGRAM TO ADD 2 NUMBERS 30 INPUT X,Y 40 LET S=X+Y 50 PRINT S 60 END
User enters RUN	RUN
Computer types ?	?389, 877
User enters 2 numbers	
Computer types answer	1266
Program ends.	DONE

Figure 1-21. Printout of program listing and run.

Control Instructions

Using this BASIC program, if you wanted to add two more numbers, you would have to start over and run (execute) the program from the beginning again. We could, however, make it add numbers over and over again, by writing in a control instruction like this:

55 GO TO 30

This instruction directs the computer to go back to statement 30, so that it keeps repeating the process step over and over. What will the computer do now? Examine the following printout. By inserting the control instruction:

55 GO TO 30

we cause the computer to return to statement 30 after it prints each sum to request two more input values. There is only one problem: we haven't programmed in a way for the computer to

```
10 REM   THIS IS AN EXAMPLE: A
20 REM   PROGRAM TO ADD 2 NUMBERS
30 INPUT X,Y
40 LET S=X+Y
50 PRINT S
55 GO TO 30
60 END
```

Figure 1-22. BASIC program with control instruction.

You load this program through the tape reader

```
10 REM   THIS IS AN EXAMPLE: A
20 REM   A PROGRAM TO ADD 2 NUMBERS
30 INPUT X,Y
40 LET S=X+Y
50 PRINT S
55 GO TO 30
60 END
```

You type RUN → RUN

Computer types ? → ?

You type 2 numbers → 389,877

Computer types sum → 1266

Above sequence repeated

```
?79,15
94
?0,0
0
?
```

Figure 1-23. Printout listing and run with control statement GO TO 30 added.

stop—this is an "endless loop." Inserting another control instruction, as in the example in Figure 1-24, below, will solve the problem.

```
10 REM   THIS IS AN EXAMPLE: A
20 REM   PROGRAM TO ADD 2 NUMBERS
30 INPUT X,Y
35 IF X=0 THEN 60
40 LET S=X+Y
50 PRINT S
55 GO TO 30
60 END
```

Figure 1-24. IF-THEN control instruction in BASIC.

The control instruction:

35 IF X = 0 THEN 60

allows you to stop the computer at any time by entering a zero as the input value for *X*. (When you do this, you must also enter a number—any number—for *Y*, since the program expects an input of *two* numbers: *X*, *Y*.) When 0 (zero) is entered as the first input number, the program will go to statement 60 and stop.

Now, examine the printout below:

Because of statement 35,

you can enter zero to stop the program.

```
10 REM   THIS IS AN EXAMPLE: A
20 REM   PROGRAM TO ADD 2 NUMBERS
30 INPUT X,Y
35 IF X=0 THEN 60
40 LET S=X+Y
50 PRINT S
55 GO TO 30
60 END

RUN

?389,877
1266
?79,15
94
?0,0
DONE
```

Figure 1–25. Printout of program with control instruction to stop the program.

Dress Up Features

Computers can output data of two types—numbers and letters. This makes it possible for a programmer to "dress up" the output with labels and messages, as shown in the printout for Figure 1–26.

This ends our brief look at the elements of computer programs. The examples we used were in the BASIC language, but the same elements make up programs in any language. With the knowledge of the three fundamental elements, you can now examine a program in almost any language and get a rough idea of its purpose:

1. Instructions concerning the *input* of data.
2. Instructions on how the data will be *processed*.
3. Instructions concerning the *output* of data.

Keep in mind that control instructions will help you find your way through the program, and programmers' remarks will help you interpret the program.

Inserting this instruction produces

```
10 REM  THIS IS AN EXAMPLE: A
20 REM  PROGRAM TO ADD 2 NUMBERS
25 PRINT "TYPE 2 NUMBERS SEPARATED BY A COMMA";
30 INPUT X,Y
35 IF X=0 THEN 60
40 LET S=X+Y
50 PRINT "THE SUM IS"; S
55 GO TO 25
60 END


RUN

TYPE 2 NUMBERS SEPARATED BY A COMMA? 389,877
THE SUM IS 1266
TYPE 2 NUMBERS DEPARATED BY A COMMA? 79,15
THE SUM IS 94
TYPE 2 NUMBERS SEPARATED BY A COMMA? 0,0

DONE
```

this output

Figure 1–26. "Dressed up" printout.

Purpose: find area of a rectangle, etc.

Input: appropriate measures.

Areas are output in lines 75, 125, 145, 165, 185, and 205.

```
20  PRINT "DO YOU WISH TO FIND THE AREA OF A RECTANGLE, PARALLELOGRAM,"
25  PRINT "SQUARE, TRIANGLE, TRAPEZOID, OR RHOMBUS?"
30  PRINT "TYPE A 1,2,3,4,5, OR 6"
35  INPUT N
40  GO TO N OF 115,135,155,175,195,65
65  PRINT "RHOMBUS - FORMULA IS A=D1*D2/2 - WHAT ARE THE DIAGONALS!"
70  INPUT D1,D2
75  PRINT "THE AREA OF THE RHOMBUS IS ";D1*D2/2;"SQ.UNITS"
80  PRINT
85  PRINT
90  PRINT "DO YOU WISH TO CONTINUE?"
95  PRINT "IF YES TYPE A 1, IF NO TYPE A 0."
100 INPUT M
105 IF M=1 THEN 30
110 GO TO 215
115 PRINT "RECTANGLE - FORMULA IS A=BH - WHAT IS THE BASE AND HEIGHT?"
120 INPUT B,H
125 PRINT "THE AREA OF THE RECTANGLE IS";B*H;"SQ. UNITS"
130 GO TO 80
135 PRINT "PARALLELOGRAM-FORMULA IS A=BH-WHAT IS THE BASE AND HEIGHT?"
140 INPUT B,H
145 PRINT "THE AREA OF THE PARALLELOGRAM IS";B*H;"SQ. UNITS".
150 GO TO 80
155 PRINT "SQUARE-FORMULA IS A=S*S- WHAT IS THE LENGTH OF A SIDE?"
160 INPUT S
165 PRINT "THE AREA OF THE SQUARE IS";S*S;"SQ. UNITS"
170 GO TO 80
175 PRINT "TRIANGLE - FORMULA IS A=1/2BH - WHAT IS THE BASE AND HEIGHT"
180 INPUT B,H
185 PRINT "THE AREA OF THE TRIANGLE IS";1/2*B*H;"SQ. UNITS"
190 GO TO 80
195 PRINT "TRAPEZOID-FORMULA IS A=1/2H(B1+B2)-ENTER HEIGHT AND 2 BASES"
200 INPUT H,B1,B2
205 PRINT "THE AREA OF A TRAPEZOID IS"1/2*H*(B1+B2);"SQ. UNITS"
210 GO TO 80
215 PRINT "IT WAS FUN CALCULATING THE AREAS FOR YOU. COME BACK AND"
216 PRINT "SEE ME AGAIN."
220 END
```

Figure 1–27. QUAD 9–sample BASIC program listing.

Samples to Interpret

Figures 1–27 and 1–28 are examples of two BASIC program listings similar to the kind of listings you may run across when looking for programs to use in your classes. To the left of the listings, we have made a rough determination of the program's purpose and other features. For practice, you may want to cover the marginal notes and try to interpret the listings on your own.

WHERE IS THE PROGRAM?

When using a computer program, you may wonder about several things: Where is the program (instructions) that makes all of this happen? What happens to that program when I sign off the terminal? If I make a mistake typing in a response will it damage or destroy the program?

Original in Storage, Copy at Work

The answer to the first question is that all programs are usually stored in the computer's permanent storage area, located either inside the CPU as a part of working storage or, more frequently, in a separate auxiliary storage device nearby. When you call the computer and ask for a certain program that is stored in it, a copy of that program is automatically taken from permanent storage and transferred into the working storage area set aside for you in the CPU. The original program stored in permanent storage is left there. This copying is very much like making

Purpose: shows equipment depreciation by 3 methods.

Values to be input: original cost, life of item, scrap value

Output headings

Formulas for computations

Operations for organizing and printing outputs

```
DEPREC

60  PRINT "THIS PROGRAM SHOWS HOW A PIECE OF CAPITAL EQUIPMENT"
70  PRINT "DEPRECIATES ACCORDING TO THREE COMMONLY USED DEPRECIATION"
80  PRINT "METHODS: STRAIGHT LINE, SUM OF THE DIGITS, AND DOUBLE"
90  PRINT "DECLINING."
120  PRINT "ORIGINAL COST";
130  INPUT C
140  PRINT "LIFE OF ITEM";
150  INPUT L
160  PRINT " SCRAP VALUE";
170  INPUT S
180  PRINT
190  PRINT "YEAR","STRAIGHT","SUM OF","DOUBLE"
200  PRINT " "," LINE","DIGITS","DECLINING"
210  PRINT
220  LET V=C-S
230  LET D1=V/L
240  LET Y=((L+1)/2)*L
250  LET Z=L
260  FOR X=1 TO L
270  LET D2=V*(Z/Y)
280  LET Z=Z-1
290  LET D3=2*C/L
300  LET C=C-D3
310  PRINT X,
312  LET Q=D1
314  GOSUB 400
316  LET Q=D2
318  GOSUB 400
320  LET Q=D3
322  GOSUB 400
325  PRINT
330  NEXT X
350  STOP
400  LET Q=INT(Q*100)/100
420  IF Q>100 THEN 440
430  PRINT " ";
440  IF Q>10 THEN 460
450  PRINT " ";
460  PRINT "$"Q,
490  RETURN
999  END
```

Figure 1–28. DEPREC—sample program listing in BASIC, with comments.

a photocopy of a page from a library book and then using the photocopy to write notes and comments on. When the photocopy is used, the original (in the book on the shelf) is not affected. If you are using a time-sharing system, it is possible that a number of different people are using copies of the same program at the same time. The interaction with the copy of a program in working storage is illustrated in Figure 1-29.

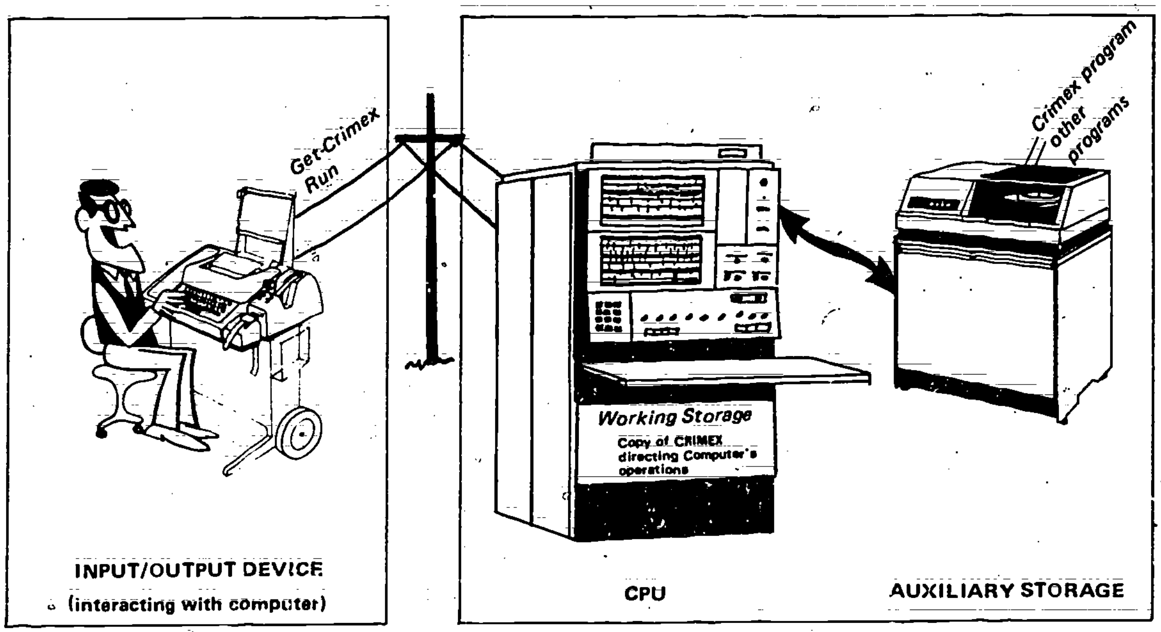


Figure 1-29. Interacting with a copy of a program in the computer's working storage.

When you sign off and leave the terminal, the copy you have been using is erased, without affecting the program in permanent storage or the other copies being used. It is possible to erase a program from the permanent storage area, but this requires special instructions written expressly for that purpose and hence it cannot be done by accident.

It is useful to think of the computer as having a library of programs. Some computers are rather small and can work with only one or two programs at one time. Others are large and can have many programs operating simultaneously. The number and size of the programs a computer can handle at any given time is often a way of determining the size of the computer.

Typing Errors

What about our last question concerning the effects of making mistakes in typing? When you make a typing mistake in your response, you will not destroy the program, so don't be afraid when you type in your responses.

II

INSTRUCTIONAL USES OF THE COMPUTER

INTRODUCTION: THE VERSATILE COMPUTER

The computer has been harnessed to perform a number of different tasks in modern society; one observer has identified over 2,000. Today, computer use is commonplace in all sectors of society.

In the last two decades, the computer has been found to be as versatile and useful a tool in education as it has in other fields. School administrators have found many ways to simplify their jobs. One of their most complex and time-consuming tasks, class scheduling, has now been computerized in most districts. Dozens of other administrative chores are made easier or taken over completely by the computer—particularly those having to do with record-keeping and clerical tasks.

The most sophisticated administrative users have begun to see the computer as a management tool, providing instant information to help them make better decisions. During salary negotiations, for example, a computer can quickly provide new salary schedules for each plan under consideration, project the costs over the next few years, and show the effect on the rest of the school budget—all while the negotiating session is still going on. As districts are faced with the necessity of cutting costs, computers can help educational managers do enrollment projections as the basis for more realistic budgeting and planning.

Many school districts across the nation are implementing computerized total management information systems, including financial, personnel/payroll, and student accounting systems. These systems are linked to function as an integrated, readily accessible management information system (MIS).

And what about teaching? This has been the last area to use computers in significant ways. Nevertheless, in the last few years a large variety of instructional applications have been introduced, and the computer has swiftly become a useful tool for both teacher and student. According to a recent study by American Institutes for Research, the number of secondary schools using the computer for instructional purposes doubled between 1970 and 1975 (from 13 to 27 percent) and will double again (to 51 percent) by 1984.

You are probably familiar with some of the growing possibilities of computer-based instruction in schools today, ranging from remedial drill work to exciting computer-enhanced discovery experiences. The overview presented in the following pages is designed to give you a brief introduction to 12 of the most common computer applications in the modern instructional setting, including some of the newest uses in guidance and instructional materials generation. While the 12 uses of the computer discussed here do not exhaust the available approaches to computers in this continually developing and changing field, they focus on the most fundamental and frequently encountered applications today.

As a means of systematizing this presentation, these 12 applications have been grouped into five basic categories, centering on the use of the computer as:

1. An *instructor/teacher;*
2. A *laboratory;*
3. A *calculator;*
4. An *object* of instruction;
5. An instructor's/teacher's *aide.*

This grouping is schematized in Figure 2–1 below.

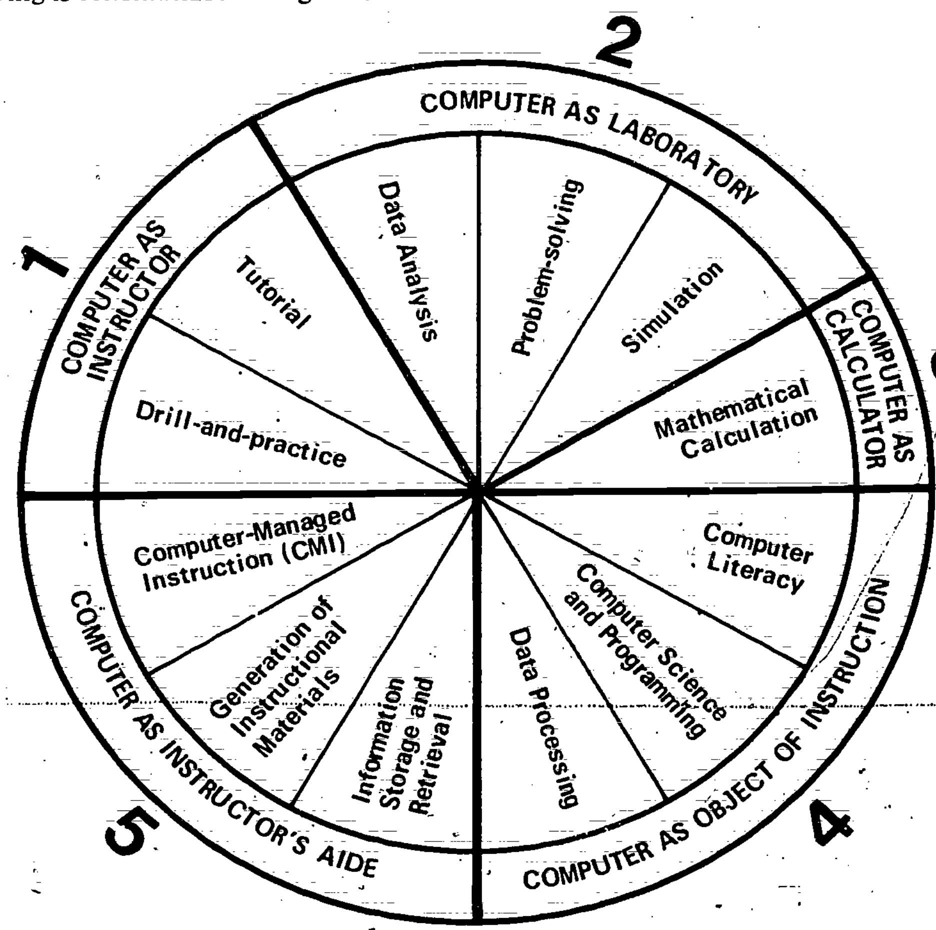


Figure 2–1. Diagram of instructional uses of the computer.

While reading over the scheme in Figure 2–1, it is important to keep in mind three things. First, to date there is no universally accepted single way of classifying instructional modes of computer use. Our present scheme represents just one way of structuring it. Second, the scheme excludes a number of less frequently encountered applications as well as the many uses which are in various stages of development at present. Third, since the computer's versatility allows it to play several roles at once, the helpful distinctions made by our diagram among general kinds of uses is quite limited; any actual instructional module (unit, application, course, lesson) may utilize the computer in a number of different, overlapping ways and hence fall into two or more of the twelve sectors at once. Keeping its limitations in mind, however, you should find the scheme in Figure 2–1 helpful for thinking and talking about current computer uses. In the following pages, each of the 12 uses in categories 1–5 is described in turn and examples of printouts or related materials are given to illustrate the special features of each use.

THE COMPUTER AS INSTRUCTOR

In a very limited sense, the computer can be programmed to function as a teacher; typically, it can provide drill, answer questions, provide examples, pose problems, dole out helpful hints, demonstrate how to do something, and so on. While it may indeed be unlikely that computers will ever replace human teachers, computer programs have been developed to perform some of the more limited and routinized teaching tasks, freeing the instructor for the more creative and conceptual tasks requiring the capacities that only the teacher possesses.

The two most common ways in which computers are used as instructors are to provide *drill*, and to function as a *tutor*.

The Computer as Drillmaster

The term "drill and practice" correctly implies that the computer can act as a drillmaster. Students first develop a skill or acquire some factual knowledge—without the computer. Perhaps they learn material from a lecture and chalkboard, a tutor, programmed instruction, a textbook, videotape, or some other means. They then use the computer to review and practice these skills and concepts.

In several disciplines—notably arithmetic, reading, spelling, history, languages, and the sciences—there are specific facts to be learned or skills to be developed. This type of learning is facilitated by practice and reinforcement, which may take hours and hours of teacher time. The use of the computer for reinforced learning frees the teacher from this task. It also permits pacing to an individual student's growth needs.

A typical drill interaction between a grade school student and a computer is shown in the printout in Figure 2–2.

This example illustrates how the computer presents a question or problem and the student types an answer. The computer compares the answer with the expected response. If it is correct, a new problem is presented, providing a subtle but positive reinforcement for giving the correct answer. If it is incorrect, the student is given another chance, and after a few incorrect tries will be given the correct answer, as is illustrated in the example in Figure 2–3.

Most drill and practice programs have essentially this same basic format: a problem is presented, the student responds, and the response is compared with the expected response. If it is correct, the student goes on to the new material; if incorrect, he is given another chance.

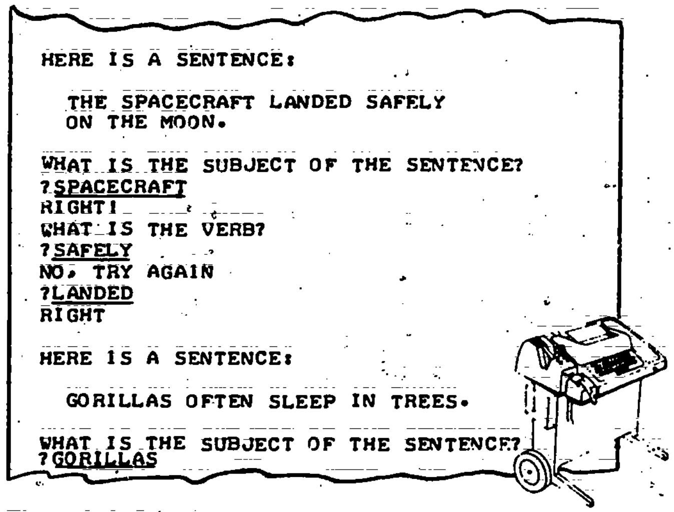


Figure 2-2. Sample drill interaction.*

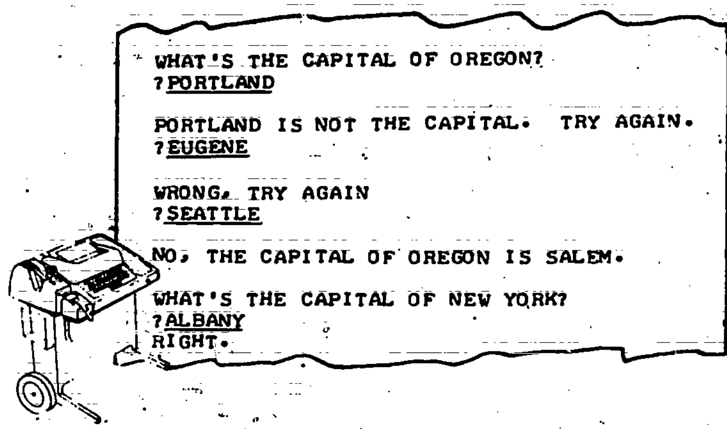


Figure 2-3. Sample interaction in which answer is given.

Many modifications are possible. For example, the program can easily be changed to provide more positive reinforcement for correct answers, as in the example shown in Figure 2-4.

If a student is having difficulty with a set of problems, he can be branched to an easier set, or to remedial material for a particular concept. In addition, wrong answers can be analyzed with a printed message telling the student what concepts need review, such as is shown in Figure 2-5.

One significant advantage of using the computer as drillmaster is that it can keep detailed records of student performance, both group and individual, and provide a variety of reports for the teacher to use in planning each student's learning experiences. Also, a student can move at his own pace through the material, spending more time with troublesome concepts or proceeding quickly through areas already mastered. Figures 2-6 and 2-7 illustrate the range of subjects for which computer drill may be appropriate.

It is important to remember that the computer when used for drill and practice *does not*

*So that the reader can easily identify the user entries on sample printouts, we have underlined them. In actual interactions, however, such underlining does not occur.

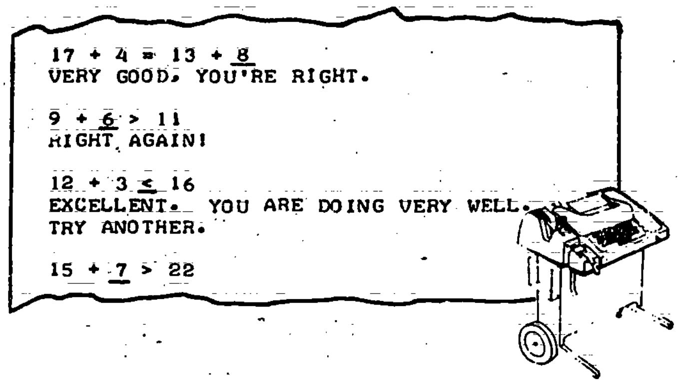


Figure 2–4. Sample drill providing reinforcement for correct answers.

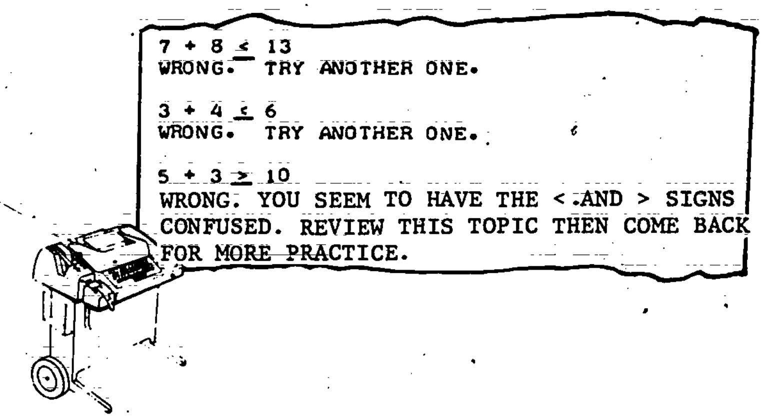


Figure 2–5. Sample drill providing guidance in concepts to review.

teach, but merely functions as a tool for practicing what the teacher is teaching. If the computer program also "teaches"—that is, presents material designed to instruct—it can more appropriately be called a *tutorial* program.

The Computer as Tutor

Tutorial programs allow a livelier interaction between student and computer than is usually provided in drill and practice units. Ideally, in the tutorial role, the computer teaches by the Socratic method: it presents factual information, then engages the student in a question-and-answer dialogue. The student may interact with the information in various contexts and applications, using and experimenting with the principles he or she has learned. Typical tutorial interaction is shown on the sample printouts in Figures 2–8 and 2–9.

Tutorial programs may be problem-oriented, that is aimed at helping students develop problem-solving or critical thinking skills in a back-and-forth exchange with the computer. Look over the example of a problem-oriented tutorial lesson in simple logic in Figure 2–9.

Diagnostic and remedial programs may also be incorporated in the tutorial so that, depending on a student's responses, he or she may be branched to the appropriate sequence.

```
RUN
MATH03
WHAT YEAR, BLOCK, AND LEVEL DO YOU WANT?6,24,5

HELLO JACK. WE HOPE YOU ENJOY TODAY'S PROBLEMS.

M 6245

********* HERE WE GO !!!!! *********

6 X 3 = ___ 18

 18 / 6 = 3

1/ 6 OF 18 = 3

4 X 4 = 16

16 / 4 = 4

1/4 OF 16 = 4

5 X 4 = 20

 20 / 5 = 4

1/5 OF 20 = 4

 7 X 4 = 28

28 / 7 = _____
DONE
```

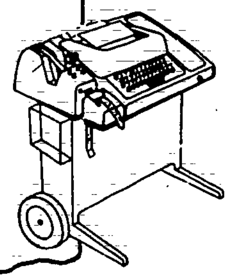


Figure 2-6. Sample interaction with drill and practice program MATH03, available from Hewlett-Packard.

```
GUTEN TAG.  ICH HEISSE HANS.  WIE HEISSEN SIE?
?MARK
IF YOU WANT TO TRANSLATE ENGLISH TO GERMAN TYPE ENGLISCH. IF
YOU WANT TO TRANSLATE GERMAN TO ENGLISH TYPE DEUTSCH
?ENGLISCH
WIEVIELE WOERTER WOLLEN SIE GEBEN? TIPPEN SIE DIE NUMMER
?2
TIPPEN SIE DAS DEUTSCHE WORT FUER THE RIVER
?DAS FLUSS
DAS GEHT NICHT
SIE HABEN DAS FALSCHE WORT GEGEBEN
TIPPEN SIE DAS DEUTSCHE WORT FUER THE RIVER
?DER FLUSS
WUNDERBAR
TIPPEN SIE DAS DEUTSCHE WORT FUER THE UNCLE
?DER ONKEL
GROSSARTIG
WENN SIE WEITERMACHEN WOLLEN, TIPPEN SIE MEHR :SONST NEIN
?MEHR
IF YOU WANT TO TRANSLATE ENGLISH TO GERMAN TYPE ENGLISCH. IF
YOU WANT TO TRANSLATE GERMAN TO ENGLISH TYPE DEUTSCH
?DEUTSCH
WIEVIELE WOERTER WOLLEN SIE GEBEN? TIPPEN SIE DIE NUMMER
?2
```

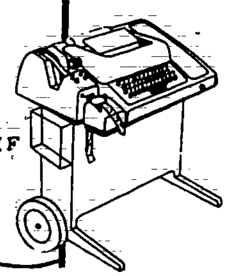


Figure 2-7. Sample interaction with German language drill program GER2, available from Project SOLO.

```
**ALL POINTS ON THE EQUATOR ARE 0 DEGREES LAT.**

TO MAKE OUR EARTH GRID WE NEED MORE E-W LINES
CROSSING THE N-S LINE.  THESE LINES ARE  CALLED
LINES OF LATITUDE (LAT.)
PLEASE WATCH THE FOLLOWING ILLUSTRATION WHILE
I MARK A SERIES OF POINTS ALL 1 DEGREE SOUTH
 OF THE EQUATOR.

                N POLE (90 DEGREES NORTH)
               • • •
            •         •
          •             •
        •                 •

       •                   •
    E  •-------EQUATOR-------•  W
       •                   •

      A •.................• B
          •             •
            •         •

               • • •
                S POLE (90 DEGREES SOUTH)

MENTALLY CONNECT THE DOTS TO FORM
A LINE OF SOUTH LATITUDE.  DONT BE CONFUSED
BECAUSE LINE AB IS AN E-W LINE.  A LINE IS
ONLY A ROW OF POINTS.  ALL THESE POINTS ARE
SOUTH OF THE EQUATOR.  THEREFORE IT IS SOUTH LAT.

A GOOD EARTH GRID WOULD HAVE--LINES OF LAT.

A. 90
B.180
C.WANT TO SIGN OFF
? B
RIGHT.
90 LINES OF S LAT. AND 90 LINES OF N LAT.
```

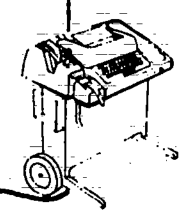


Figure 2-8. Sample interaction with a geography tutorial program available from Minnesota Educational Computing Consortium.

```
HERE IS A STATEMENT:

  IF IT RAINS THEN LYLE WILL STAY HOME.

FIRST, ASSUME IT DOES NOT RAIN.  CAN YOU CONCLUDE
THAT LYLE WILL NOT STAY HOME?
?NO
RIGHT.

ASSUMING THAT LYLE STAYS HOME, CAN YOU CONCLUDE
IT RAINS?
?YES
NO, YOUR CONCLUSION IS INVALID: YOU DENIED THE
ANTECEDENT.
```

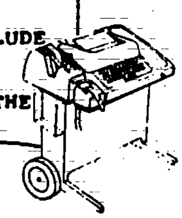
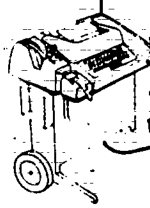


Figure 2-9. Sample logic tutorial.

While a drill and practice program usually is programmed to discriminate only right responses from wrong ones, some tutorials are programmed to deal with any conceivable response, including slang words and swearing. Here is a sample printout from an interactive tutorial program which can deal with a wide range of student responses.



```
RECALL FROM OUR PREVIOUS LESSON THAT GEORGE II RULED
GREAT BRITAIN FROM 1727 TO 1760, AND THAT DURING HIS
REIGN BRITAIN HAD A SERIES OF WHIG PRIME MINISTERS.

WHO WAS THE FIRST PRIME MINISTER TO SERVE UNDER GEORGE
II?  LET'S SEE--WAS IT COMPTON?
NO--COMPTON WAS THE  S E C O N D  PRIME MINISTER TO
SERVE UNDER GEORGE II; PLEASE TRY AGAIN.

WHO WAS THE FIRST PRIME MINISTER TO SERVE UNDER GEORGE
II?  WAS IT WILSON, MAYBE?
NOW STOP PLAYING AROUND; NO ONE BY THAT NAME EVER
SERVED AS PRIME MINISTER IN GEORGE II'S TIME. PLEASE
TRY AGAIN.

WHO WAS THE FIRST PRIME MINISTER TO SERVE UNDER GEORGE
II?  //HINT
THE MAN I'M THINKING OF ALSO SERVED AS PRIME MINISTER
UNDER GEORGE I. NOW TRY AGAIN.
```

Figure 2-10. A tutorial interaction which handles a wide range of student responses—program available from Minnesota Educational Computing Consortium.

Notice the flexibility of computer responses in the tutorial role.

Since a tutorial assumes the burden of the teaching task, you will notice a variety of styles and formats in tutorial programs. Some incorporate the elements of learning theory and are carefully designed; others reflect a less sophisticated approach to teaching. An example of a rather complete and complex tutorial interaction is shown in Figure 2–11.

THE COMPUTER AS LABORATORY

The computer can act as a laboratory in a wide variety of instructional situations by providing an environment in which the student can examine information, construct and examine models, and conduct experiments. Typically, the computer presents some portion of the real world in a way that allows the student to learn by experimentation and observation.

There are three different approaches to using the computer as a laboratory: the data analysis approach, the problem-solving approach, and the simulation approach.

Briefly, in the *data analysis approach*, the real world is represented by a set of data (a "data base") describing some aspect of the real world. The student runs experiments with the data and tests hypotheses about real-world relationships by comparing the different categories.

In the *problem-solving approach*, the student learns about some aspect of the real world by

```
>RUN

LATTTT        14:28:21     7 JUL 71

HELLO AGAIN.  WHO ARE YOU.  TYPE YOUR NAME
AFTER THE QUESTION MARK.
? JADE
YOU MAY FIND THIS PROGRAM TAKES YOU LONGER
THAN TEN MINUTES TO OPERATE.  WATCH YOUR TIME.
WHEN YOU SEE TIME IS NEARLY UP, ANSWER THE
NEXT QUESTION WITH THE SIGN OFF CHOICE.
COME BACK ANOTHER TIME AND TRY IT AGAIN.

**EVERY CIRCLE HAS 360 DEGREES**

WHAT ELSE DO YOU WANT TO KNOW ABOUT CIRCLES.

A. RELATED WORDS FOR GEOGRAPHY
B. DEFINITIONS OF A CIRCLE AND A SPHERE
C. NUMBER OF CIRCLES ON A SPHERE
D. NOTHING RIGHT NOW
E. WANT TO SIGN OFF
? C

LAT1          14:30:04     7 JUL 71

**NUMBER OF CIRCLES ON A SPHERE**

BECAUSE CIRCLES, JUST LIKE STRAIGHT LINES,
ARE A ROW OR SERIES OF POINTS, THEY ARE
IMAGINARY UNTIL WE DRAW THEM.  WE CAN IMAGINE
ANY NUMBER OF CIRCLES ON A SPHERE AND THEN
DRAW THEM.

REMEMBER:

A.THERE REALLY ARE NO CIRCLES DRAWN ON THE

  PLANET EARTH.  THEY ARE DRAWN ONLY ON THE

  GLOBE OR CERTAIN MAPS.

B.A CIRCLE ON THE PLANET MAY HAVE ANY RADIUS

  YOU CHOOSE.

C.A CIRCLE WITH THE SAME RADIUS AS THE SPHERE
  IT IS DRAWN ON IS CALLED A GREAT CIRCLE.

  1.IT IS THE LARGEST CIRCLE THAT CAN BE
    DRAWN ON A SPHERE.

  2.IT ALWAYS CUTS THE PLANET INTO TWO
    EQUAL PARTS (HEMISPHERES)

  3.IT IS THE SHORTEST DISTANCE BETWEEN TWO
    POINTS ON THE SURFACE OF THE SPHERE.
```

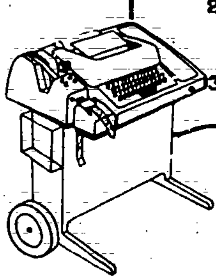


Figure 2-11. Sample geography tutorial available from Minnesota Educational Computing Consortium.

writing or using a computer program to solve a problem. This may involve a data base, a model, or a mathematical equation. The approach has traditionally been used in mathematics, science, and business, but it may also be used in the social sciences or the humanities. Problem-solving programs are often structured as games, especially by student programmers.

In the *simulation approach*, the real world is represented by a model which is believed to behave like some portion of the real world. The interaction may be either a straightforward simulation or a game. A simulation game allows each student to take partial or total control of one side of the situation being modeled and to make the decisions and negotiate or revise strategies or other elements at work. In this way, the student can investigate the impact of his decisions. Interacting with a simulation/game, a student can typically test a strategy, experience the implications of his choices, and gain insight into the factors involved and their importance.

Data analysis, problem-solving, and simulation all present the student with an environment in which he can learn by experimentation. The chief difference between the data base and the simulation approach is in how the real world is presented: the data base represents it as data collected from the real world, and the simulation represents it as a model which generates data similar to data which might be collected in the real world. The problem-solving approach may use a data base or a simulation-type or mathematical model. Frequently in the problem-solving approach, students write the program in order to master the algorithm involved in the solution.

The Computer in Data Analysis

One of the ways the computer has proven itself an invaluable aid in nearly every field to which it has been applied is in the analysis of data. With a computer, human beings can store information, selectively retrieve it, sort it, and analyze it in a fraction of the time it would take using manual methods.

In the instructional world, this data analysis capability has particular relevance to teachers of social studies and the sciences, where students learn to collect data, analyze them, and evaluate the results. The method of collecting data will vary from the survyes and polls of the social scientist to the observations in a science laboratory, but the need for efficient analysis and evaluation is the same. Since teachers' instructional objectives often emphasize interpretation of results rather than tedious and time-consuming computation of data, the availability of a computer for the computation makes it possible for teachers to assign projects and experiments previously very time-consuming or even impossible.

Take, for example, a social studies class which undertakes a special project: to conduct a political preference poll of every student in a 900-member student body. The class wishes to store a data base in the computer—a pool of information about the political preferences of the students. The information can then be analyzed in whatever way the class desires. The questionnaire might include such items as the student's grade, age, sex, religion, party preference, candidate preference, mother's party affiliation, father's party affiliation, and so on. Once these data are coded and stored, the class members can retrieve them in any form they wish. They can ask for a report comparing the party preferences of grades 9 and 10 and grades 11 and 12; or the candidates preferred by girls as compared to boys; or a breakdown of candidate preferences by grade, showing the votes for each candidate as a number and a percentage of the total vote. An analysis could also be made of the apparent influence of a parent's political party affiliation on the party preference of the student.

Depending on the sophistication of the students and the objectives of the teacher, the computer can be programmed to perform various statistical tests on the data, allowing students formally to test hypotheses about their student body.

A data analysis program called SAP (Survey Analysis Package),* developed for the Huntington Two unit, performs statistical calculations which allow students in grades 10 to 12 to examine large quanities of survey data. The program description from the Teacher Manual for the unit is reprinted in Figure 2-12.

II. DESCRIPTION OF THE PROGRAM

The program in this unit will do statistical calculations useful for examining and analyzing large quantities of survey data. The amount of raw data that the user may enter into the program will depend on the storage capacity of the computer. (Please consult someone familiar with the computer you are using to help you determine the number of respondents you can store data for.) The number of questions that can be stored for each respondent is also dependent on storage capacity. In the rare case where your computer will not accept " Ø" as a valid subscript, we have included an option in the program to avoid the problem.

After the user has entered the data, he can have the program do any of the following:

1) Compute the mean and standard deviation for any variable. (With OPTIONS 1 through 5 a Recode OPTION is included.)

2) Construct a table of observed frequencies for any two variables, compute chi square, degrees of freedom (df), and allow the user to calculate the correlation coefficient gamma if desired.

3) Construct a table of observed frequencies by row percentages for any two variables.

4) Construct a table of observed frequencies by column percentages for any two variables.

5) Construct a table of expected frequencies for any two variables.

6) Stop the program.

If a minus sign (-) is typed before the option number, the program will use previously stored data to make its calculations. For example, if the user selects OPTION 2, he will be asked to name two variables; then the program will make the calculations and print the results using these variables. If the same two variables are to be used with another option, the user should type " - " and then the option number. This will cause the same two variables to be used in the calculation and will save considerable time by not having the machine re-read the data from data line storage. We suggest that you become familiar with this option, since it will save you time.

Figure 2-12. Sample program description from the "Survey Analysis Package" unit by Huntington Two Computer Project.

*"Survey Analysis Package" of the Huntington Two Computer Project is available from Digital Equipment Corporation, Maynard, Massachusetts.

Data analysis programs like SAP are also used as a base for hypothesis-testing exercises, using data organized by the teacher or another curriculum developer. As an example, read over the excerpts in Figure 2–13 taken from the "Analyzing Crime" unit, which uses the data analysis system INQUIR as a base for formulating and testing hypotheses about crime. The Analyzing Crime unit is available from Hewlett-Packard.

Crime: Its Frequency and Occurance

We know that big cities are the most troubled by crime, especially what has come to be known as "street crime." We know also that certain areas of cities have more of a crime problem than other neighborhoods. What has escaped us so far is why certain people commit crimes, or alternatively, why certain areas have more crimes than others. These exercises won't answer this question for you because, in part, no one has an answer. You will, however, uncover something about the characteristics of neighborhoods and the occurrence of three kinds of crime. The data are from the New York Times and describe the 71 police precincts in the city. Each variable has been recoded into four categories where one represents the highest category and four the lowest. Appendix A contains the full list of variables.

Exercise 1

One variable frequently used to describe crime rates is race. Discover for yourself the relationship between race and crime by crosstabulating the proportion of blacks by each of the three crimes. What do you conclude?

Exercise 2

Now investigate another ethnic minority, Spanish speaking residents or their immediate families. Do this by crosstabulating the proportion of Spanish speaking residents by each of the three crimes. Do you notice any differences? To what might you attribute this?

A second school of thought argues that crime is basically a result of poverty, i.e., poor people commit more crimes because (1) they profit by doing this and (2) being unemployed they have nothing else

Figure 2–13. Excerpt from the Hewlett-Packard unit, "Analyzing Crime."

Data analysis is as important in scientific endeavors as the laboratory experiments which generate the data. In the science laboratory, the available time and money usually restrict the individual student to making one run through an experiment; but because duplicate runs are usually necessary before any degree of confidence can be placed in a result, the one-run limit on student experiments too often results in students' erroneous conclusions.

To circumvent this problem, teachers and/or students may perform some data analysis on classroom results to correct for these errors. For example, to separate acceptable experimental results from unacceptable ones for a class, the teacher might have all students record their results on one graph; the teacher would then eliminate by observation those results which do not seem to fit the main group.

A less arbitrary approach is to use Chauvenet's criterion for statistical rejection of deviant data. This involves calculating the lower and upper bounds from some criterion level based on the distribution of the data. The data items that do not meet the criterion are discarded, and the criterion level is recalculated. The process is continued until all remaining items meet the test; the average and standard deviation is then calculated for these items. A histogram of the remaining items is made, and the expected Gaussian distribution is calculated from the average, the standard deviation, and the sample size. The data histogram is compared with the concomitant Gaussian distribution, and difference and chi-squared fit are calculated. On the basis of this analysis the acceptability of each experimental result can be accurately determined, permitting valid experimental conclusions to be drawn.

Exactly this kind of data analysis is made possible by such a computer program as LABSTA, developed by Project REACT.* In the example in Figure 2–14, the results 55 students obtained from an experiment with magnesium and hydrochloric acid are analyzed. First, the 55 items of data (volume of hydrogen gas as calculated by students) were entered; then LABSTA performed the necessary data analysis. The essential parts of the LABSTA run are shown in Figure 2–14.

The Computer in Problem-Solving

In the area of instructional problem-solving, the computer can play two different roles: (1) it can be used directly to solve problems too complex or time-consuming for hand solutions to be practiced; and (2) it can be used as a vehicle for teaching problem-solving skills through computer programming. These two roles involve very different materials, interactions, and results. So we will consider them one at a time.

Solving Complex Problems

The computer can provide great enrichment to the curriculum when its capacity for flexible problem-solving is used effectively. In this role, it is not a mere calculator; rather, it is a tool for solving complex problems and making large amounts of otherwise inaccessible information available to students.

Unlike tutorial or drill-and-practice use, problem-solving by the computer provides little in the way of original instruction or correction for the student. Interaction with problem-solving programs is usually characterized by the computer's requesting the user to enter data needed to solve the problem. The computer performs all needed manipulations and provides the user with the solution. For example, in the following interaction (Figure 2–15) the student enters basic population characteristics requested by the computer and the computer prints out the requested population projection.

An important feature of problem-solving programs is their capacity to solve many problems of the same kind quickly, one after the other. This means that students can explore particular problems in considerable depth and/or breadth. To facilitate this use, problem-solving programs may offer several options for solving related problems. Notice those offered at the end of the sample run shown in Figure 2–16 on page 47.

*Relevant Educational Applications of Computer Technology, Northwest Regional Educational Laboratory, Portland, Oregon, 1971.

```
THESE VALUES LOOK OUT OF PLACE AND SO ARE REJECTED.
VALUE          AVERAGE        DEVIATION      PERMITTED DEVIATION

PASS 1
 49.3           22.6113        26.6887        10.8898

PASS 2
 13.7           22.0981        -8.39808       4.30281
 15.9           22.0981        -6.19808       4.30281

PASS 3
 19.4           22.39          -2.99          1.65225
 25             22.39           2.61          1.65225

THERE ARE 48   VALUES IN THE FULL CLEAN SET

AVERAGE        SIGMA

  22.3979        0.2578
     :            :                :              :

LOWER          UPPER
BOUND          BOUND          HISTOGRAM BARS

 21.7999        22.0400       ****
 22.0400        22.2800       ********
 22.2800        22.5200       ************************
 22.5200        22.7600       ********
 22.7600        23.0001       ****


EXPECTED HISTOGRAM

COUNT       EXPECT       DIF          CHI2

  4          3.49       -0.51          0.07
  8         11.73        3.73          1.19
 24         17.56       -6.44          2.36
  8         11.73        3.73          1.19
  4          3.49       -0.51          0.07

SUMS OF VALUES:

 48         48.00       14.92          4.88
```

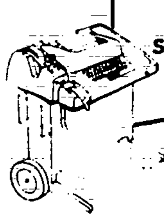


Figure 2-14. Sample LABSTA data analysis program printout.

For instructional use, problem-solving programs must rely on information and problems provided to the student by textbooks, by laboratory and classroom presentations and materials, or by student materials specially designed for use with the program. An example is given on page 48 of a student worksheet designed to be used with the problem-solving program FACTRI, developed by Project REACT. The worksheet provides the student with a problem involving the

```
WHAT FERTILITY WILL BE STABLE? 2.06
HOW MANY DECADES UNTIL FERTILITY REACHES 2.06        ? 3

DO YOU ASSUME STANDARD BIRTH DISTRIBUTION (1=YES,0=NO)? 1

DO YOU ASSUME STANDARD SEX RATIO (1=YES,0=NO)? 1

DO YOU ASSUME STANDARD MORTALITY (1=YES,0=NO)? 1

DO YOU ASSUME STANDARD POPULATION (1=YES,0=NO)? 1

REPORT:1)SHORT 2)LONG 3)GRAPH 4)CHANGE ASSUMPTIONS 5)END? 1

YEAR 1970       POP=  204.8        MILLION      FERTILITY 2.4
YEAR FOR NEXT REPORT? 2000

REPORT:? 3

YEAR 2000       POP=  264.2        MILLION      FERTILITY 2.06
                            PCT. TOTAL POP.
                 0.........5.........10.........15........20
 0    - 4        .         *
 5    - 9        .          *
 10   - 14       .           *
 15   - 19       .           *
 20   - 24       .          *
 25   - 29       .         *
 30   - 34       .        *
 35   - 39       .          *
 40   - 44       .          *
```

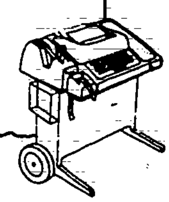


Figure 2-15. Sample problem-solving interaction with the Huntington Two Project Program USPOP.

factorization of a trinomial. Instructions are then given for using FACTRI to help solve the problem.

Detailed directions for running the program as well as a sample run usually accompany computer-oriented student materials. Program procedures and a sample run from the FACTRI unit are illustrated in Figures 2-17b and c.

A combination of effective guide materials and problem-solving programs can provide more than just problem solutions. If a student is guided in generating a substantial range of solutions to similar problems, he or she can be further encouraged to learn, by "discovery," new patterns, rules, and implications from the data. Another worksheet designed for use with program FACTRI illustrates this potential.

Before we consider how computers are used to teach problem solving, look over the two sample printouts of interactions which use the computer directly to solve problems (Figures 2-15 and 2-16).

Student Written Programs

The second role of the computer in problem solving is characterized by programs which students write themselves, in order to solve a problem. This is probably the most popular instructional use of the computer to date, since typically fewer curriculum materials are required and prewritten programs are not needed.

```
ENTER VALUES FOR A,B, AND C ON ONE LINE,
USING COMMAS TO SEPARATE THEM.
?-3,2,5

ENTER VALUES FOR X(1), X(2), AND I
ON ONE LINE, USING COMMAS TO SEPARATE THEM.
?-10,5,1

FUNCTION:  Y = -3 * X↑2 +2 * X +5

      X          Y

     -10        -315
     -9         -256
     -8         -203
     -7         -156
     -6         -115
     -5         -80
     -4         -51
     -3         -28
     -2         -11
     -1          0
      0          5.
      1          4
      2         -3
      3         -16
      4         -35
      5.        -60

TO ENTER A NEW INTERVAL, TYPE 1.
TO ENTER A NEW FUNCTION, TYPE 2.
ANY OTHER NUMBER WILL STOP THE PROGRAM.
?1
```

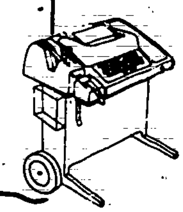


Figure 2-16. Sample interaction with program offering options.

The first discipline to make use of the computer in this way was mathematics. It has been said that in mathematics there are three levels of understanding: the first comes when you hear a concept explained; the second, when you explain the concept to someone else; and the deepest comes when you explain the concept to a computer. Programming a computer demands an attention to logic and precision as well as a complete understanding of the concept being programmed.

Some typical materials for problem-solving through computer programming are shown in Figure 2-21. In this example, part of the student material on the problem-solving methodology is illustrated, along with a sample of how students are guided into writing the problem-solving program. In this case, the student must write a program to solve some elementary matrix problems. To do this, he will need a thorough understanding of the matrix operations involved, as well as the ability to write some programs in BASIC.

Today, many disciplines make use of program writing as an effective means of developing problem-solving skills. A portion of a typical problem-solving application related to air pollution is shown in Figure 2-22. This unit, called "The Automobile and Air Pollution," is designed to allow students to focus on a problem simply at first and then to use more and more complex problem-solving skills as they work their way through the series of exercises.

Student-written problem-solving computer games are a popular learning device. Typical are hide-and-seek games, in which the user tries to guess where a player is hiding in a mathematically numbered grid. For example, in a 10 by 10 grid, the computer might "hide" the player at the point (4, 7) as shown in Figure 2-23. The user enters different grid points by number and, as hints are given about how far off the last guess was, the guess must be refined in the effort to

Name______________________

Class/Section______________________

Program Used: FACTRI

Application 2: Aiding Calculations

At times you may be working on problems involving trinomials of the form $ax^2 + bx + c$ in which complicated factorizations arise. In these instances, FACTRI can be used to obtain the factorizations.

Example

At your desk: Work out the following problem until you have an expression which needs to be factored.

> A rectangular swimming pool has a length which is twice its width. The pool is to be surrounded by a boardwalk 4 feet wide. The total area covered is to be 12474 square feet. Find the dimensions of the pool.

You may want to draw a diagram to help you visualize the problem situation.

At the terminal: Use FACTRI to obtain the factorization you need to solve the problem.

At your desk: Solve the problem, using the space below for your work.

Figure 2-17a. Sample student materials for Project REACT's mathematics unit based on program FACTRI. Above, student worksheet.

PROGRAM PROCEDURES AND NOTES (FACTRI)

> The program FACTRI gives the factorizations for trinomials of the form $ax^2 + bx + c$ and identifies those trinomials given that are not factorable.

1. Make on-line connections and call up the program FACTRI according to the instructions appropriate for your machine. (Remember to return the carriage after each of your entries so the computer can accept the next entry or proceed with the program.)

2. The program will begin by asking you to indicate how many expressions you wish to factor. (Enter the appropriate number.)

3. The program will next ask you to enter the coefficients for each expression you want factored. For each expression, enter the three coefficients a, b, and c on one line, using commas to separate them.

   Example: To factor the two expressions $x^2 - 9$ and $x^2 + 5x + 6$, you would make the responses underlined below:

```
HOW MANY EXPRESSIONS DO YOU WISH TO FACTOR?
?2

FOR EACH TRINOMIAL ENTER VALUES FOR A, B AND C,
USING COMMAS TO SEPARATE THEM.
?1,0,-9
?1,5,6
```

(Note: Remember the three values must be entered for each expression, even if one of the values is 0.)

After the run has been completed, the program will ask if you want to factor additional trinomials. If so, type "YES" and proceed as in steps 2 – 4. If not, type "NO" and the program will conclude.

Figure 2-17b. Sample program instructions from FACTRI student materials.

PROGRAM SAMPLE RUN (FACTRI)

```
THIS PROGRAM WILL FACTOR A TRINOMIAL OF THE FORM
   A*X↑2 + B*X + C

HOW MANY EXPRESSIONS DO YOU WISH TO FACTOR?
?5

FOR EACH TRINOMIAL ENTER VALUES FOR A, B AND C,
USING COMMAS TO SEPARATE THEM.
?1,0,-4
?8,-6,24
?36,-72,9
?-50,-80,10
?8,3,4

THE FACTORS OF X↑2 - 4 ARE:
     (X + 2) (X - 2)

THE FACTORS OF 8*X↑2 - 8*X + 24 ARE:
     8 (X↑2 - 4*X + 18)

THE FACTORS OF 36*X↑2 - 72*X + 9 ARE:
     9 (4*X↑2 - 8*X + 1)

THE FACTORS OF -50*X↑2 - 80*X + 10 ARE:
     -10 (5*X↑2 + 8*X - 1)

8*X↑2 + 3*X + 4 CAN NOT BE FACTORED.

DO YOU WANT TO FACTOR MORE TRINOMIALS?
?NO
```

Figure 2-17c. Sample program run from FACTRI student materials.

-5-

Name ____________________

Class/Section ____________________

Program Used: FACTRI

Application 3: Generating Research Data

It would be nice if you could "just look" at a trinomial expression like $x^2 - 9xr - 360r^2$ and, with very little trouble, know whether or not it can be factored. Is there some relationship between the coefficients a, b, and c, or any two of these, which would be a major clue to the "factorability" of trinomial expressions? FACTRI can help you to research questions like this.

Example

At the terminal: Using program FACTRI, input the coefficients a, b, and c of the following trinomials:

	Trinomial	Enter for a, b, and c
(1)	$x^2 + 4x - 12$	1, 4, -12
(2)	$x^2 - 5x - 36$	1, -5, -36
(3)	$6x^2 - 7x - 2$	6, -7, -2
(4)	$6x^2 + 11x + 7$	6, 11, 7
(5)	$2x^2 + 9x - 18$	2, 9, -18
(6)	$9x^2 - 9x - 4$	9, -9, -4
(7)	$6x^2 + 22x + 14$	6, 22, 14
(8)	$x^2 + x - 12$	1, 1, -12
(9)	$x^2 - 4x - 14$	1, -4, -14
(10)	$3x^2 + 19 + 6$	3, 19, 6

At your desk: Using the data generated above, examine the coefficients a, b, and c for the expressions which can be factored and for those which cannot be factored.

(1) Are there any relationships between the trinomial coefficients which indicate whether or not a trinomial can be factored? Record your hunches here and test them out by using FACTRI

Figure 2-18. Sample worksheet illustrating "discovery" guidance.

```
                    FINANCIAL PROBLEMS

THIS PROGRAM SOLVES THREE TYPES OF PROBLEMS:

     (1) INTEREST ON INSTALLMENT BUYING
     (2) PAYMENTS ON LONG TERM LOAN
     (3) BALANCE OF A SAVINGS ACCOUNT

WHICH PROBLEM WOULD YOU LIKE TO WORK WITH (TYPE 1, 2 OR 3)?1

*****

THIS SECTION WILL DETERMINE THE ACTUAL INTEREST YOU PAY
WHEN YOU PURCHASE SOMETHING ON CREDIT.

WHAT IS THE CASH PRICE OF THE ARTICLE ($)?88.99
            DOWN PAYMENT ($)?10
            NUMBER OF PAYMENTS EXCLUDING THE DOWN PAYMENT?18
            NUMBER OF PAYMENTS PER MONTH?1
            AMOUNT PER PAYMENT ($)?4.85


THE RATE OF INTEREST CHARGED WAS 5.69        PERCENT.


*****

WOULD YOU LIKE TO RUN THE PROGRAM AGAIN (1-YES, 0-NO)?1
WHICH PROBLEM WOULD YOU LIKE TO WORK WITH (TYPE 1, 2 OR 3)?2

*****

THIS SECTION WILL DETERMINE PAYMENTS FOR A LONG TERM LOAN.

WHAT IS THE AMOUNT BORROWED ($)?3000
            INTEREST CHARGED (%)?8
            INTERVAL BETWEEN PAYMENTS (MONTHS)?1
            TERM OF THE LOAN (YEARS)?1

DO YOU WISH TO SEE THE TOTALS ONLY - INSTEAD OF THE ENTIRE
TABLE - (1-YES, 0-NO)?0

           OUTSTANDING
           PRINCIPAL AT                         PRINCIPAL
           BEGINNING        INTEREST DUE AT     REPAID AT
PERIOD     OF PERIOD        END OF PERIOD       END OF PERIOD

 1          3000             20                  240.97
 2          2759.03          18.39               242.58
 3          2516.45          16.78               244.19
 4          2272.26          15.15               245.82
 5          2026.44          13.51               247.46
 6          1778.98          11.86               249.11
 7          1529.87          10.2                250.77
 8          1279.1           8.53                252.44
 9          1026.66          6.84                254.13
10          772.53           5.15                255.82
11          516.71           3.44                257.53
12          259.18           1.73                259.24
                            ---------------     -------------
TOTALS                       131.58              3000

YOUR MONTHLY PAYMENT IS $ 260.97      AND TOTALS $ 3131.58
```

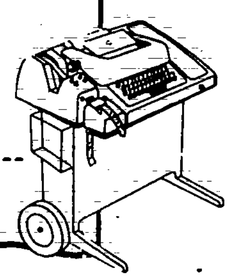


Figure 2-19. Sample interaction with business problem-solving program BANK, available from Minnesota Educational Computing Consortium.

PROGRAM SAMPLE RUNS (FRADEC)

Run 1

```
THIS PROGRAM WILL COMPUTE THE DECIMAL REPRESENTATIONS
OF RATIONAL NUMBERS THAT ARE IN FRACTIONAL FORM.

HOW MANY RATIONAL NUMBERS DO YOU WANT TO CONVERT?
?3

HOW MANY DECIMAL PLACES DO YOU WANT?
?12

ENTER THE RATIONAL NUMBERS, ONE TO A LINE, BY TYPING
THE NUMERATOR, A COMMA, AND THE DENOMINATOR.

?3,5
?65,54
?6,7


RATIONAL       DECIMAL
NUMBER         REPRESENTATION


3 / 5  =  .6

65 / 54  =  1.203703703703

6 / 7  =  .857142857142

DO YOU WANT TO MAKE A NEW RUN?
?YES
```

Run 2

```
HOW MANY RATIONAL NUMBERS DO YOU WANT TO CONVERT?
?3

HOW MANY DECIMAL PLACES DO YOU WANT?
?102

ENTER THE RATIONAL NUMBERS, ONE TO A LINE, BY TYPING
THE NUMERATOR, A COMMA, AND THE DENOMINATOR.
?345,543
```

Figure 2-20. Sample run of a problem-solving program, FRADEC, from student materials to REACT's "Fractions to Decimals" unit.

THE AIRLINE TICKETING PROBLEM

EAGLE Airlines, a little known but ambitious "scheduled" commuter service, lists the following direct flights:

Eichelbergertown to Salladasburg

Eichelbergertown to Kingerstown

Eichelbergertown to Lackawaxen

...

... to sum all of the power matrices (1st power, 2nd power, etc.) from the 1st power to the nth power, where n is the number of cities (and therefore the number of rows (or columns) in the binary matrix A). [If you don't remember how to add matrices, work through the module "ELEMENTARY MATRIX OPERATIONS".]

In the mathematical notation this is

$$S = A^1 + A^2 + A^3 + \dots + A^{n-1} + A^n$$

where S is obtained by adding the matrices on the right. The entries in the summation matrix are the total number of paths of any length (1 to n) connecting any two towns.

PROBLEMS FOR COMPUTER SOLUTION

PROBLEM 1

Write a computer program which finds the number of different ways in which a person can fly from a given starting city to each of the other cities (including the starting city) via EAGLE Airlines.

SAMPLE SOLUTION

On page 8 we will show a flow chart describing one solution of this problem; on page 9 a listing of a BASIC program that implements this flow chart is given. This program does not use the matrix functions available in in most versons of BASIC in order to show you how a professional programmer manipulates matrices.

Figure 2-21. Sample student material from "Communication Matrices," available from Project SOLO.

EXERCISE 1 – Estimating Number of Cars

How many automobiles would you expect to find in our residential area? How many automobiles would you expect to find running at some arbitrary time? You will have to make some assumptions to reach your answer. Be sure and state these assumptions explicitly. Compare your assumptions to those of other students. Do your assumptions stand up well under close examination?

Now that you have estimated the number of cars, we will structure our first model. Let P stand for the number of cubic feet of pollutants at any time, R for the number of cubic feet of pollutants produced per hour by each car, and N for the number of cars operating at any given time. The simplest model we could construct would be

$$P_{new} = P_{old} + (R)(N). \qquad (1)$$

P_{new} is the amount of pollutants at the end of any hour. P_{old} is the amount at the end of the previous hour.

Initially, let's concentrate solely on carbon monoxide pollution. This gas is fairly stable and quite persistent. We will need some concentrations and their effects to use in the exercises. A carbon monoxide concentration of 1000 parts carbon monoxide to one million parts air (abbreviated 1000 ppm) is sufficient to produce unconsciousness in 1 hour and death in 4 hours. The maximum allowable concentration for industrial workers for an eight-hour working day is 50 ppm. Concentrations of from 25 to 50 ppm will be experienced inside an automobile moving in a heavy stream of traffic in a multilane highway or freeway.

EXERCISE 2 – A Simple Model

Write a BASIC program to compute and print out the number of cubic feet of carbon monoxide in our residential area every hour for a 24-hour period. Assume that at the beginning there is no carbon monoxide in the air. Use the number of cars running which you estimated in Exercise 1. The carbon monoxide production rate per car can be obtained from Table 2.

EXERCISE 3 – Computing Concentrations

Modify the program in Exercise 2 to print out the carbon monoxide con- ...

Figure 2-22. Sample student materials from problem-solving unit "The Automobile and Air Pollution," available from Hewlett-Packard.

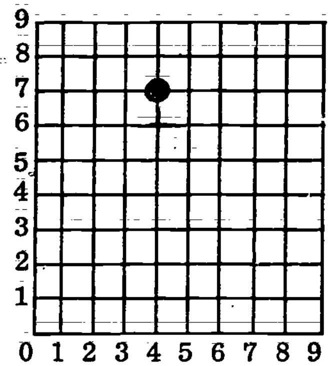


Figure 2–23. Sample grid for a hide-and-seek game.

locate the hidden player. The games can be structured very simply, as is the case with the following sample interaction, or they can become very complicated.

```
DO YOU WANT TO SEE THE RULES?YES

A PLAYER IS HIDING IN A 10   BY 10   GRID.  TRY TO
FIND HIM BY GUESSING HIS GRIDPOINT.  HOMEBASE IS
GRIDPOINT 0,0  AND A GUESS IS A PAIR OF WHOLE
NUMBERS (0 TO 9    ) SEPARATED BY A COMMA.  THE FIRST
NUMBER IS THE DISTANCE TO THE RIGHT OF HOMEBASE
AND THE SECOND NUMBER IS THE DISTANCE ABOVE THE
HOMEBASE.  FOR EXAMPLE, IF YOU THINK THE PLAYER
IS HIDING 8 UNITS TO THE RIGHT OF HOMEBASE AND
3 JNITS ABOVE HOMEBASE, THEN ENTER 8,3 AS YOUR
GUESS AND PRESS THE 'RETURN' KEY.

YOU GET 5    GUESSES.  AFTER EACH GUESS, I WILL
TELL YOU HOW FAR (IN A DIRECT LINE) YOU ARE FROM
THE PLAYER.

PLAYER IS HIDING.  YOU GET 5    GUESSES.

WHAT IS YOUR GUESS?7,6
YOU ARE 4.4         UNITS FROM THE PLAYER.

WHAT IS YOUR GUESS?5,3
YOU ARE 1     UNITS FROM THE PLAYER.

WHAT IS YOUR GUESS?4,2
YOU ARE 1     UNITS FROM THE PLAYER.

WHAT IS YOUR GUESS?5,2
YOU FOUND HIM IN 4     GUESSES!!!
```

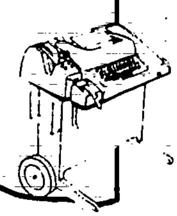


Figure 2–24. Sample interaction with a hide-and-seek game written as a step in mastering mathematical concepts.

Two types of learning are possible here. The students who *use* the program (play the game) learn that if they apply some principles about triangles they can find the hidden player faster. But students who *write* the program must not only have a clear understanding of those same principles, but must thoroughly understand and apply the "distance formula"–an important formula in algebra based on the Pythagorean theorem. The problem solved by the student who writes the program is simply this: write a program which will tell a user how far a given point (the user's guess) is from a randomly selected point (the hidden player) in a 10 by 10 grid.

Simulations and Simulation/Games

One of the most exciting and promising uses of the computer to enhance classroom instruction is for instructional simulations and games. This approach includes pure simulations and simulation/games. Initially, we can distinguish between the two in the following way.

Simulations are operating models of physical or social situations. The word *model*, when used in the context of simulation, means two things. First, it means a framework in which the reality being represented is reduced in size until it reaches manageable proportions. Second, it means a framework in which only certain aspects of the real thing are chosen for inclusion; in other words, reality is simplified.

While simulations give a simplified picture of real physical or social systems, they nevertheless attempt to replicate essential aspects of these spheres of reality so that the reality may be better understood and/or controlled. The degree to which the designer selects essential elements from reality determines how closely or completely the reality is simulated.

Games can be defined as competitive interactions among participants to achieve specified goals. Often highly motivational for students, they are usually played for entertainment and clearly identify winners and losers; success depends on skill or chance or some combination of the two. Games need not attempt to replicate real-world behavior–rules of behavior need apply to the game only.

From these two ideas–simulation to represent elements of reality and gaming to motivate–have developed a variety of powerful learning contexts commonly known as educational simulation/games, which present a real-world model within a game-playing context.

In the social studies, pure simulation as well as simulation/games can allow students to explore and discover cause-and-effect relationships, develop and try out strategies, and learn how a social, economic or political system operates. By participating in a simulation or game, students develop both a better understanding of how systems operate and a respect for the complex interplay of variables in social systems.

Following are sample interactions with programs which simulate an election system, a system of socioeconomic influences on crime rates, and a system of international balance of payments. As you read through the typical interaction shown for each, notice the kinds and number of things the user can manipulate and the kinds and ranges of the (simulated) results. Also notice which programs appear to be the more formal simulations and which have incorporated some characteristics of a game program.

In the sciences, computer simulations can allow for simulating experiments that could not be conducted otherwise because there would be danger or expense, or the required equipment or facilities were impossible to obtain, or the time scales involved were too short or too long for analysis in the school setting. Two typical examples of computer-enhanced simulations in the sciences follow. The first is of a simulated experiment with radioactive decay. The second is a computer-simulated genetics experiment which would involve facilities normally difficult to get as well as a time span that is too long for the typical school laboratory.

```
                              ELECT2

               HISTORICAL ELECTIONS - 20TH CENTURY

NEED INSTRUCTIONS?YES


YOUR GOAL WILL BE TO CHOOSE THE OPTIMUM STRATEGY FOR
CANDIDATES IN AN HISTORICAL ELECTION.  EACH CANDIDATE'S
STRATEGY VECTOR CONSISTS OF 3 NUMBERS.  THE FIRST REPRESENTS
THE AMOUNT OF EMPHASIS TO BE PLACED ON THE CANDIDATE'S
IMAGE, THE SECOND REPRESENTS THE AMOUNT OF EMPHASIS ON PARTY
AFFILIATION, AND THE THIRD IS THE AMOUNT OF EMPHASIS ON
CAMPAIGN ISSUES.  EACH OF THESE NUMBERS IS BETWEEN 10 AND 80,
WITH A HIGHER NUMBER REPRESENTING MORE EMPHASIS.  THE TOTAL
OF EACH STRATEGY VECTOR MUST EQUAL 100!

THE COMPUTER WILL FIRST ASK 'ELECTION YEAR?'
CHOOSE THE YEAR FROM THE FOLLOWING LIST:
  1920  1928  1932  1948  1952  1960  1968

ELECTION YEAR?1948


CANDIDATE       PARTY
TRUMAN          DEMOCRAT
DEWEY           REPUBLICAN

TRUMAN?30,55,15
DEWEY?50,20,30


THE RESULT OF YOUR STRATEGY IS:
TRUMAN          51.25 %
DEWEY           48.75 %


THE VOTE FOR THE TWO MAJOR CANDIDATES IN THE ACTUAL ELECTION WAS:
TRUMAN          52.40 %
DEWEY           47.60 %
```

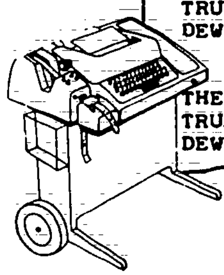


Figure 2-25. Sample interaction with simulation program ELECT2, available from Digital Equipment Corporation.

CRIMEX

DO YOU WANT INSTRUCTIONS?YES

THIS PROGRAM SIMULATES CRIME IN A TYPICAL CITY WITH FIVE PRECINCTS. YOU MAY CHANGE CONDITIONS IN THE PRECINCTS TO TRY TO REDUCE CRIME RATES. TO STOP THE PROGRAM, ENTER THE WORD DONE IN ANSWER TO ANY QUESTION.

DO YOU WANT TO OMIT UNEXPECTED EVENTS?YES

INITIAL CRIME RATE (PER 100,000 PEOPLE)

	PRECINCT 1	2	3	4	5	CITY
VIOLENT CRIMES	69	45	31	30	29	41.0
NON-VIOLENT CRIMES	294	256	214	208	199	234.5

DO YOU WANT THE CURRENT PROFILE OF OTHER CONDITIONS?YES

	1	2	3	4	5	CITY
1 - PER CAPITA INCOME	2600	3700	3950	4130	4400	3756
2 - UNEMPLOYMENT RATE	13.0	11.0	8.0	6.0	4.0	8.[illegible]
3 - GENERAL UPKEEP	1	2	2	3	4	2.[illegible]
4 - POPULATION	179000	128000	100000	84000	61000	552000
5 - NUMBER OF POLICE	358	282	240	202	159	1241
6 - TRAINING LEVEL	2	3	3	3	4	3.0
7 - PRE-TRIAL BAIL	80	80	80	90	95	85.[illegible]
8 - SENTENCING	12	9	7	3	3	6.8

DO YOU WANT TO CHANGE THE VALUE OF ANY OF THE CONDITIONS?YES

ENTER PRECINCT, ITEM NUMBER AND NEW VALUE. FOR EXAMPLE IF YOU WANT TO INCREASE THE NUMBER OF POLICE IN PRECINCT 3 TO A TOTAL OF 240, YOU WOULD ENTER 3,5,240 AFTER THE QUESTION MARK IS PRINTED. WHEN YOU HAVE NO MORE CHANGES TO MAKE, ENTER 0,0,0

?1,1,2850
?2,1,4000
?1,2,12
?2,2,9.5
?2,2,10
?0,0,0

CRIME RATE - END OF YEAR

WHICH PRECINCTS?ALL

	PRECINCT 1	2	3	4	5	CITY
VIOLENT CRIMES	73	47	35	33	[illegible]	44.4
NON-VIOLENT CRIMES	[illegible]	261	[illegible]	212	202	239.5

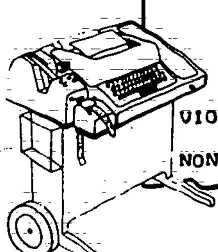


Figure 2-26. Sample interaction with [illegible] [illegible]ation, CRIMEX, developed by [illegible] Northwest Regional Educational Laboratory

```
BALPAY

DO YOU WANT A DESCRIPTION OF THIS GAME?YES

THIS COMPUTER PROGRAM IS A DECISION MAKING GAME.
YOUR PART IN THE GAME IS THAT OF DECISION MAKER
FOR THE COUNTRY.  ASSUME THAT THE COUNTRY IS CURRENTLY
IN A VERY POOR BALANCE OF PAYMENTS POSITION.  YOUR
OBJECTIVE IS TO MAKE DECISIONS THAT WILL GIVE THE
COUNTRY A HEALTHY BALANCE OF PAYMENTS POSITION WITHIN  4      YEARS.

BALANCE STATEMENT IN MILLIONS OF DOLLARS - YEAR 0

INFLOWS

EXPORTS        GOODS & SERVICES      $ 19 073
EXPENDITURES FOREIGN TOURIST         $  2 153
INCOME       INVESTMENT              $    902

NET INFLOW   $ 22128

OUTFLOWS

IMPORTS       GOODS & SERVICES       $ 18 271
EXPENDITURES TOURISTS ABROAD         $  1 610
FOREIGN AID & MILITARY AID           $  4 745
INVESTMENTS FOREIGN                  $  2 375

NET OUTFLOW   $ 27001

NET SURPLUS OR DEFICIT/NET INFLOW   -.23

AS YOU CAN SEE FOR THE STATEMENT ABOVE, OUTFLOWS
EXCEED INFLOWS BY A SUBSTANTIAL AMOUNT:  YOUR COUNTRY
IS IN A SERIOUS DEFICIT POSITION.  SHOULD THE DEFICIT
CONTINUE FOR LONG, YOUR COUNTRY'S GOLD RESERVES WOULD
BE DEPLETED.  ALSO TOO MUCH OF YOUR COUNTRY'S RESOURCES
AND PRODUCTION ARE PROBABLY BEING SENT OUT OF THE COUNTRY

FOR THE PURPOSES OF THIS GAME, A 'HEALTHY' BALANCE OF
PAYMENTS POSITION IS DEFINED AS ONE IN WHICH THE VALUE
OF THE RATIO, 'NET SURPLUS OF DEFICIT/NET INFLOW' IS
BETWEEN THE VALUES OF -.03        AND  .03        .  NOTICE
THAT THE CURRENT VALUE OF THIS RATIO IS -.23

THERE HAS BEEN A REVOLUTION IN PERU.  THE NEW
GOVERNMENT HAS TAKEN OVER ALL OF THEY CONTROLLED
BY FOREIGN INVESTORS.  ALSO HARSH RESTRICTIONS HAVE BEEN
IMPOSED ON IMPORTS FROM YOUR COUNTRY.

ENTER YOUR POLICY DECISIONS FOR YEAR  1

WHAT PERCENT CHANGE DO YOU WISH
IN TAXES ON INVESTMENTS ABROAD?3
IN THE TARIFF RATE?4.5
IN GOVERNMENT SPENDING ABROAD?-4
IN THE PRIME INTEREST RATE?.5

BALANCE STATEMENT - YEAR 1

INFLOWS

EXPORTS        $ 18 209
EXPENDITURES   $  2 133
INCOME         $    866

NET INFLOW   $ 21208


OUTFLOWS

IMPORTS        $ 15 502
EXPENDITURES   $  1 555
FOREIGN AID    $  4 646
INVESTMENTS    $  1 888

NET OUTFLOW    $ 23591
$ DIFFERENCE   -2383
% DIFFERENCE   -.12
```

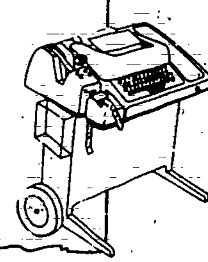


Figure 2-27. Sample interaction with economics simulation available from Tecnica Education Corp.

DECAY2—Radioactive Decay Simulation

```
DECAY2

DO YOU WANT INSTRUCTIONS (1=YES, 0=NO) : ?1
  THIS PROGRAM WILL DO THE FOLLOWING:
       CHOICE 1 - CALCULATES HALF-LIFE FROM TWO READINGS
                  ON A GEIGER COUNTER.
       CHOICE 2 - CALCULATES HOW MUCH OF A RADIOACTIVE SAMPLE
                  WILL REMAIN AFTER SOME GIVEN AMOUNT OF TIME
       CHOICE 3 - PRINTS OUT A TABLE SHOWING MASS OF SAMPLE
                  PVS. TIME OR NO. OF PARTICLES VS. TIME.
                   (GRAPH OPTIONAL) NOTE: FOR THE TABLE YOU
                   MUST INPUT TOTAL TIME AND TIME INCREMENT.
                   EXAMPLE: IF TOTAL TIME=100 AND TIME
                   INCREMENT=10, THEN TIME IN THE TABLE WILL
                   BE 10,20,30,..............100.
        CHOICE 4 - END OF PROGRAM

                   NOTE: IN ANY ONE PROBLEM,TIME MUST
                   ALWAYS BE INPUTED IN THE SAME UNITS
                   OF MEASURE (IE: SECS.,MINS.,ETC.)
**********

WHAT IS YOUR CHOICE?1

 WHAT IS THE INITIAL READING ON THE GEIGER COUNTER,
 THE SECOND READING, AND THE TIME BETWEEN READINGS.
?1500,3000,36

INITIAL READING= 1500    SECOND READING 3000    TIME= 36
HALF-LIFE= 35.9976

**********

WHAT IS YOUR CHOICE?3

DO YOU WANT TO WORK WITH PARTICLES OR MASS? (ANSWER 1 FOR
 PARTICLES OR 2 FOR MASS) ?1

 WHAT IS THE HALF-LIFE, INITIAL NUMBER OF PARTICLES IN THE
 SAMPLE, TOTAL ELAPSED TIME FOR DECAY, AND THE
 INCREMENT OF ELAPSED TIME?10,6.02E23,100,10

HALF-LIFE= 10   INITIAL NO. OF PARTICLES= 6.02000E+23
TOTAL TIME= 100  INCREMENT= 10

 TIME           PARTICLES      PART. LOSS      TOTAL PART. LOSS
 ----           ---------      ----- ----      ----- ----- ----

 0              6.02000E+23    0               0
 10             3.01014E+23    3.00986E+23     3.00986E+23
 20             1.50514E+23    1.50500E+23     4.51486E+23
 .                   .              .               .
 .                   .              .               .
 90             1.17628E+21    1.17617E+21     6.00824E+23
 100            5.88168E+20    5.88112E+20     6.01412E+23

 DO YOU WANT THE ABOVE DATA GRAPHED? (1-YES, 0-NO)?1
                                   MASS (OR PARTICLES) REMAINING

                0
 6.02000E+23
TIME            I---------I---------I---------I---------I---------I
 0              I                                                 *
 10             I                        *
 20             I               *
 30             I       *
 40             I  *
                I*
```

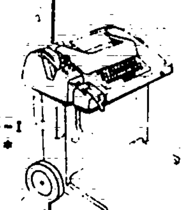


Figure 2-28. Sample interaction with science simulation, DECAY2, available from Tecnica Education Corp.

```
CHOOSE:
  1. COMPLETE DOMINANCE
  2. ALLOWANCE FOR OTHER GENETIC INTERACTIONS
?2

ENTER THE NUMBER OF THE OPTION FOR WHICH YOU WISH
TO SEE THE EXPECTED DISTRIBUTION OF OFFSPRING
     1. THE NTH GENERATION
     2. EVERY NTH GENERATION
?1

WHAT IS THE VALUE OF N?1

WITH WHAT SPECIES WILL YOU BE WORKING?MAN

HOW MANY PHENOTYPE CHARACTERISTICS WILL BE INVOLVED?4


WHAT IS CHACTERISTIC 1     ?HAIR COLOR

WHAT ONE WORD SPECIFIES THE CHARACTERISTIC
   1) FOR THE FIRST PARENT?BLACK
   2) FOR THE SECOND PARENT?BLOND
                 .                 .                          .
                 .                 .                          .
                 .                 .                          .
ENTER THE TRANSMISSION PROBABILITIES UNDER THE
APPROPRIATE COLUMN HEADINGS (YOU MUST SPACE
AND PLACE A COMMA BETWEEN ENTRIES ON A LINE)

                CHARACTERISTIC  1
                  HAIR COLOR

  CROSS                CHARACTERISTIC OF OFFSPRING

                                 BLACK       BLOND       OTHER
BLACK        BLOND              ?.75,        .15,         .10
               .                  .                        .
               .                  .                        .
               .                  .                        .
WHAT IS THE EXPECTED NUMBER OF OFFSPRING
FOR EACH PAIR OF PARENTS?12

            THE EXPECTED DISTRIBUTION OF OFFSPRING AMONG
            THE POSSIBLE COMBINATIONS OF PHENOTYPE TRAITS
                      FOR THE SPECIES, MAN


                         GENERATION 1

                     NEW OFFSPRING= 12

COMB.                    PHENOTYPE TRAITS                 EXP.  EXP.
NUMB.                     OF OFFSPRING                    PER.  NUM.

       HAIR COLOR   EYE COLOR   NOSE       FEET

  1    BLACK        BROWN       STRAIGHT   LARGE          23     3
  2    BLACK        BROWN       STRAIGHT   SMALL           9     1
  3    BLACK        BROWN       STRAIGHT   OTHER           3     0
  4    BLACK        BROWN       PUG        LARGE           6     1
  5    BLACK        BROWN       PUG        SMALL           2     0
  6    BLACK        BROWN       PUG        OTHER           0     0
```

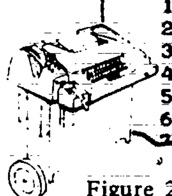


Figure 2-29. Sample interaction with genetics simulation, GENET, available from Tecnica Education Corp.

THE COMPUTER AS CALCULATOR

Because of the computer's popularly recognized speed and accuracy in performing complex calculations, it would seem that its most obvious role in education would be as a calculator, just as it is in science, research, industry, and so on. In instruction, however, this use is really quite limited; its calculation capacities are typically coupled with other capabilities that result in the kinds of uses we have been discussing, which may calculate but also do much else.

At this point, the question might arise: But isn't the use of the computer as a problem solver the same thing as its use as a calculator? It is true that the uses are similar; but there are two basic differences which allow us to distinguish between these uses in a helpful way. First, a problem-solving program is often written by a student for the purpose of mastering the underlying algorithm or mathematical concept; in contrast, formal calculator programs are assumed to be pre-prepared or "canned" specifically for straightforward calculational purposes, especially in areas where complexity and length make hand calculations impractical. Second, pre-prepared problem-solving programs are typically tailored for specific instructional purposes and include such special features for student use as particular formats of output or options for varying input and outputs to achieve instructional goals; calculational programs, on the other hand, are normally as brief and economical as possible in their input and output features. Purely calculational programs, then, can be defined as those which simply calculate solutions to mathematical problems and are used primarily where highly complex calculations advise or require the assistance of electronic computation capabilities. As a result, in the educational setting, the use of the computer strictly as a calculator is generally restricted to mathematics, science, business, and statistics courses, where highly complex calculations may be required which are not the central focus of the subject.

As an example of a calculator program, look at the printout in Figure 2–30 and compare it

```
N =?26
DATA =?100,98,97,94,94,91,88,85,84,83,83,83,79,77,74,74,74,74,68,65
??61,53,50,46,44,25
MEAN = 74.7692
VAR = 357.465
SD = 18.9067

DONE


RUN
[illegible]

N =?26
DATA =?91,90,90,88,87,83,82,82,82,74,72,70,68,68,68,62,5[illegible],58,55,54
??5[illegible],48,48,43,38,19
MEAN = 66.5385
VAR = 346.738
SD = 18.6209
```

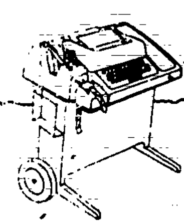


Figure 2-30. Excerpt from an interaction using the computer strictly as a calculator.

to the problem-solving program printouts in Figures 2–17c and 2–19. These show that a calculator program is typically a bare-answer producer, while a pre-prepared problem-solving program may offer instructionally advantageous options and features in addition to solutions.

Often, it may not be clear which goal was the original intent of a program. In the school situation, such a program can probably serve in either role.

The computer uses we have discussed to this point have used the computer as an active participant in the instruction process, whether to help deliver instruction or to enhance and support instruction. These participatory categories can be illustrated on our diagram as in Figure 2–31.

The remaining two categories involve the computer not as a participant but as an *object* to be studied by students and as a *tool* at the disposal of the teacher for storing and retrieving information, generating materials, and keeping records.

Let's turn our attention now to these uses of the computer in the school setting.

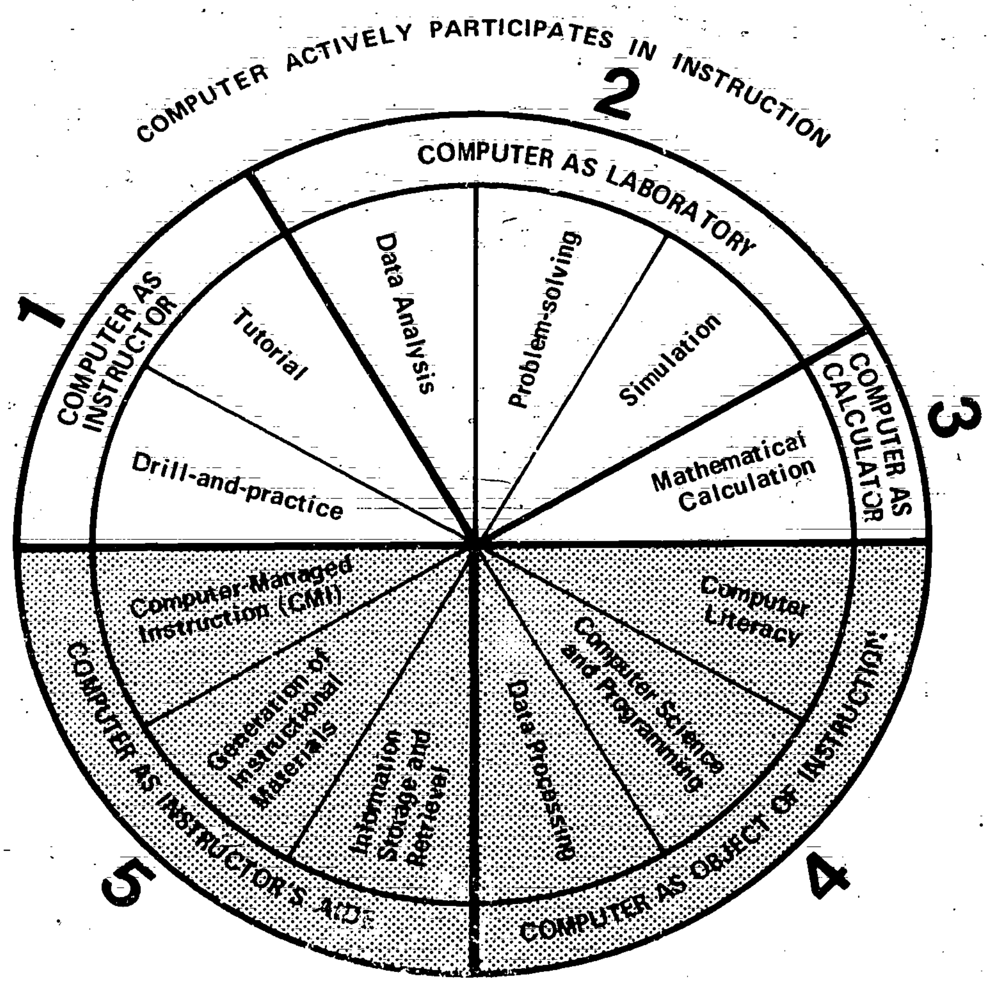


Figure 2–31. Diagram showing uses of computers which are active in instruction process.

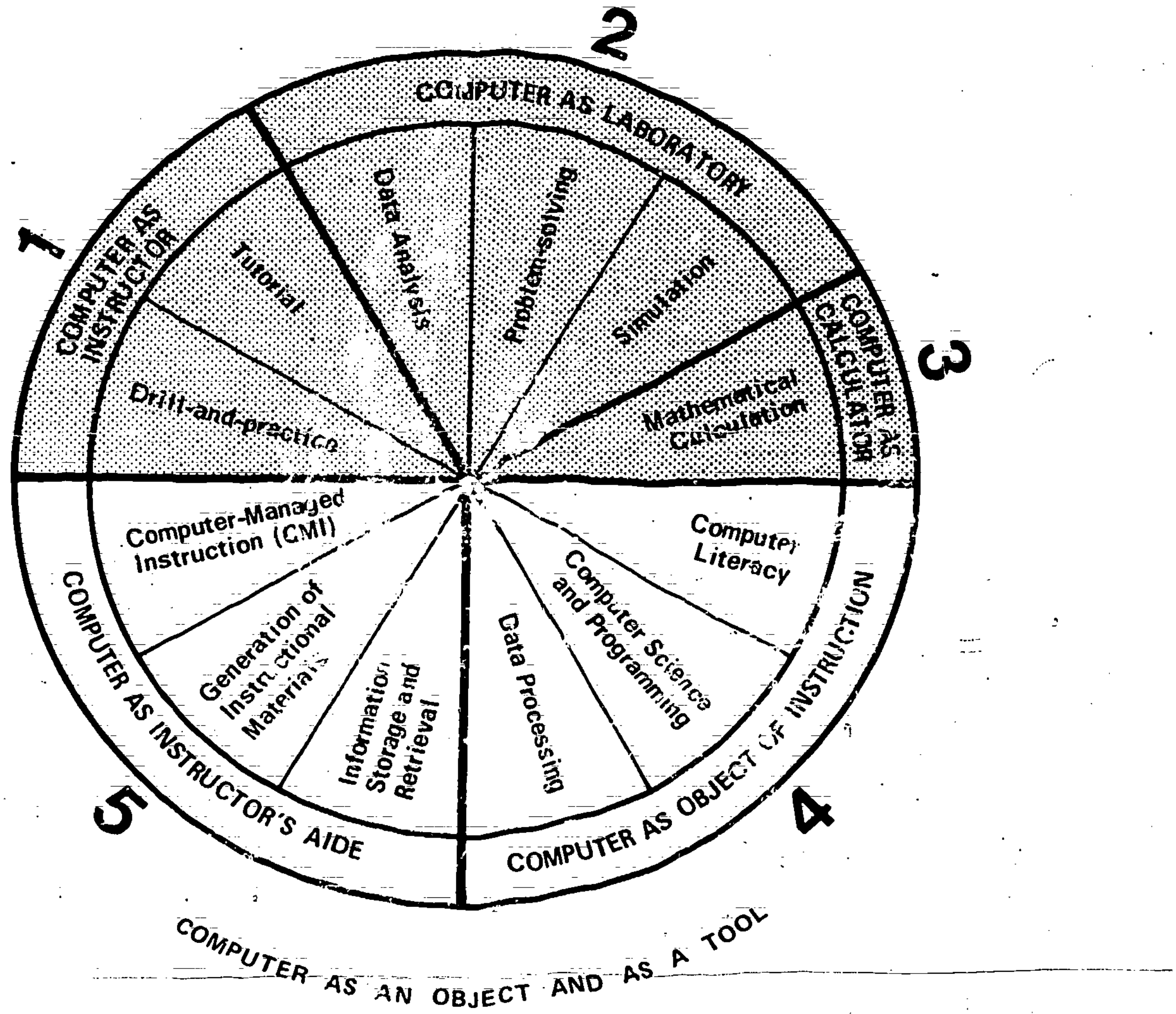


Figure 2–32. Uses of the computer as an object of instruction and teacher's tool.

THE COMPUTER AS AN OBJECT OF INSTRUCTION

When the computer is used as an object of instruction, we are teaching not *with* the computer but *about* it. When the computer made its first incursion into the classroom in the mid-1960s, it was primarily as an object which gifted mathematics and science students could learn to write programs for and run their own programs on. Today, the frequency with which computers are encountered in the daily life of our society has required a much broader exposure to this technology in schools at all levels.

Currently, there are three main approaches to the computer as an object of instruction. First, computer literacy courses deal with the nontechnical and low-technical aspects of the capabilities and limitations of computers; they usually pay special attention to the social, cultural, vocational, and educational implications of computers. Second, computer science and programming courses generally teach students how the computer works and how to program for it. Finally, data processing courses instruct students in the basics of using the computer as a processor, including such skills as preparing data for computer processing and programming for business data processing.

Computer Literacy

Of these three approaches, the first usually involves the most indirect contact with computers, though some exposure to computers at work and to computer use is ordinarily included. Computer literacy courses typically focus on the impact of the computer on society, dealing with their ability to assist in solving social and economic problems and their threat to privacy and individuality. Other areas that might be included are the nature and implications of artificial intelligence, computer-related vocational opportunities, and use or creation of computer programs in diverse fields.

Courses in this field have been called for by many different groups and individuals, since it has become apparent that there is a growing urgency to create a citizenry competent to understand the uses and problems of the technology in order to be able to employ and control it intelligently. While increasing instructional use of computer-based units such as those of computer-assisted instruction (CAI) subject the student to the power of the computer, courses in computer literacy aim directly at equipping students with the understanding needed to be masters of the technology.

Computer Science and Programming

The study of computer science and programming may be a part of the curriculum at nearly any level of education today. Courses in computer science usually include a practical introduction to the computer as a machine, giving attention to the basics of how it operates, how to run programs on it, and how to write programs. These courses may be given independently of mathematics, science, or business courses, or they may be associated with a related discipline in order to offer students more specialized and useful exposure to computer programming. Separate courses in programming languages* are becoming increasingly available as students at higher levels of education make increasing use of computers. At the college level, the separate discipline of computer science is well established and offers a complete study of the computer for future professionals as well as for majors in the growing number of disciplines which make use of computers.

Today, an increasing number of colleges and universities expects entering students to have computer programming skill along with the minimal requirements in language arts and mathematics. The trend is reflected in secondary schools, where computer science courses are increasingly available. As inexpensive microprocessors find their place in schools, more students have an opportunity to develop actual software, probably in machine or assembly language, and to learn how to control a computer at a personal level.

Data Processing

As one of the most important uses of the computer today in nearly every field, data processing has become the focus of special study offered in secondary schools and colleges as well as in career-training schools, typically in the business education curriculum. Courses ordinarily include practical training in keypunching (preparing data on a keypunch machine), computer operation, and/or programming for data processing operations. As opposed to a computer science course, programming training for data processing focuses less on the computer and its manipulation than on the business application itself. A student of computer science might develop complex software

*Several of these languages are discussed in Part I of this book.

to control the computer; a student of data processing, on the other hand, might develop a program to process accounts receivable or keep track of inventory for a hypothetical shoe store. Other students, less interested in or capable of writing programs, might learn other skills, such as how to operate a keypunch machine, a card sorter or collater, or a computer. Often, these skills can lead directly to or enhance the student's employability.

THE COMPUTER AS INSTRUCTOR'S AIDE

The computer as an instructor's aide can provide an enormous amount of assistance to the teacher by performing some of the noninstructional functions, freeing the teacher to spend more time actually teaching. The three major areas in which the computer can help are sorting and retrieving information (information handling), helping to prepare or generate instructional materials (worksheets and tests), and helping the teacher manage the instructional process.

Information Storage and Retrieval

Computers are very effective and efficient tools for storing and searching through large volumes of data for selective retrieval upon request.

Although teachers routinely consult various sources for information in the process of planning and managing instruction, until recently they have had to find the needed information in catalogues of printed and nonprinted materials, bibliographies, or student records, which has required tedious and time-consuming searching through files, libraries, and indexes. With the help of computers, teachers are beginning to enjoy the advantages of having such information accurately and instantly accessed and retrieved.

One of the major areas in which the computer is currently being used to store and selectively retrieve large volumes of information is in counseling and guidance. A number of computer-based career planning and occupational information systems have been developed and are in use to assist young people in learning about present and future educational or career opportunities and make them aware of occupations they would find personally satisfying. The computerized information system aids the overburdened counselor by encouraging students engaged in career exploration and decision making to seek vocational information on their own. Such a system not only stores and retrieves vast quantities of data, but sorts quickly through the data to find those which have certain characteristics prescribed by the student; and it brings together, compares, and relates various bits of information, all during an interactive dialogue with the student. Students may explore occupations suited to their own interests, values, and aptitudes and determine educational requirements for these occupations. Some systems allow the student to explore a variety of available options of post-secondary education and training and to ask for information on available scholarships.

One such system is the Guidance Information System (GIS), copyrighted by Time Share Corporation. It is described by the Minnesota Educational Computing Consortium as follows:

Name: GIS

Description: The Guidance Information System is a computerized information storage and retrieval system available on the MECC computer. GIS provides immediate access to information about occupations, 2-year colleges, 4-year colleges, vocational and technical schools and financial aid available in Minnesota and the nation.

Comments: The information is stored in five different data files. A brief description of each file follows:

COL4: The four-year college file (COL4) contains information on more than 1,600 colleges and universities in the U.S.A. The student may examine over 750 characteristics such as costs, financial aid, size and available majors.

COL2: The two-year college file (COL2) contains information on approximately 1,000 two-year community, technical and junior colleges in the U.S. The student may examine over 400 characteristics related to these colleges.

OCCU: The occupation file (OCCU) contains information on more than 1,200 careers. The student may examine over 400 characteristics that include educational requirements, working conditions, salary ranges, future job prospects, personal interests and aptitudes.

SCHO: The scholarship and financial aid file (SCHO) contains financial aid information from such sources as the federal government, foundations and other organizations. The Aid Programs are national in scope and require that students apply directly to the funding agency.

VOCA: The vocational and technical school file (VOCA) contains information about vocational and technical education in Minnesota.

Using the GIS Program: After obtaining the system library program (GIS) the user must choose one of the five data files to be utilized. (COL4, COL2, VOCA, OCCU, or SCHO). It is necessary to refer to a *GIS Student Study Guide* at this time in order to choose the appropriate characteristic and characteristic number to use in his search. When this number is entered along with a one-letter code the computer will respond with the number of colleges (or schools, occupations, scholarships) that meet your search requirements. If less than 25 items are left in that data file as a result of the search the user may printout those items and get additional information about them; otherwise, the search must be continued.

Another example is the Oregon Career Information System (CIS). Excerpts from a counseling session using CIS are found in Figure 2–33 on the following pages.

Materials Generation

The capacity of the computer system to act as a "smart printing press" (combining fast printing with information retrieval) has been made available to teachers for the purpose of generating instructional materials. In this mode of computer use, special materials-generation programs with access to extensive stored data allow teachers to specify the kinds of materials needed; once the teacher's needs have been entered, the computer proceeds to retrieve and print out materials to those specifications. The kinds of materials which can be printed out range from tests and games to individualized worksheets and text materials.

One example of a materials-generation system is the Computer Based Resource Unit (CBRU) developed by SUNY at Buffalo, New York, which helps teachers create individualized instruction materials. For any one of over 70 topics, the teacher may specify his objectives and the characteristics of the students. The CBRU system then searches through a bank of information and prints out appropriate instructional activities, content items, resources lists, and test items keyed to the objectives and to the learners. The system automatically includes suggestions to match the profile of learner variables for each student as well as for the class as a whole.

```
HELLO; PLEASE ENTER YOUR NAME, THEN PUNCH THE 'RETURN' KEY.
?MARILYN
HELLO MARILYN,
YOU ARE LOGGED IN TO THE CAREER INFORMATION SYSTEM;
USE IT AS MUCH AS YOU WANT.  WHEN YOU ARE THROUGH, PLEASE
TYPE IN 'STOP' SO YOU DO NOT WASTE COMPUTER TIME.

NOW, HOW DO YOU WANT TO START?
IF YOU FILLED OUT THE QUESTIONNAIRE IN YOUR HANDBOOK,
         TYPE IN: 'QUEST.'
IF THERE ARE OCCUPATIONS, EDUCATIONAL PROGRAMS,
OR SCHOOLS YOU WANT INFORMATION ABOUT,
         TYPE IN: 'INFO.'
?INFO
THREE KINDS OF INFORMATION ARE STORED IN THE COMPUTER:
- - - OCCUPATIONS:  LOCAL, STATE, AND NATIONAL LABOR MARKET
        INFORMATION THAT IS CONTINUOUSLY UPDATED.
             TYPE IN 'DESC' AND A 4-DIGIT OCCUPATION CODE
             (EXAMPLE: DESC 1684).  YOU MAY ALSO FIND OCCU-
             PATIONAL BIBLIOGRAPHIES (BIB); VISITS (VISIT);
             EXPLORER POSTS (XPLOR); AND WAYS TO PREPARE FOR
             OCCUPATIONS (PREP).

- - - PROGRAMS OF STUDY AND TRAINING: TYPICAL
        COURSEWORK AND LIST OF SCHOOLS.
             TYPE IN 'PROG' AND A 3-DIGIT PROGRAM CODE
             EXAMPLE: PROG 131

- - - SCHOOLS:  SERVICES AND COSTS OF SCHOOLS.
             TYPE IN 'SCH'

YOU CAN FIND CODE LISTS IN YOUR HANDBOOK.

?DESC 8456
WHICH AREA ARE YOU IN:
PORTLAND METROPOLITAN AREA? (TYPE IN: PORT)
MID-WILLAMETTE AREA? (TYPE IN: WILL)
LANE COUNTY? (TYPE IN: LANE)
SOUTHERN OREGON? (TYPE IN: SO)
COOS-CURRY COUNTIES? (TYPE IN: COOS)
EASTERN OREGON? (TYPE IN: EAST)
OTHER AREA? (TYPE IN: OTHER)
?LANE
 8456    ELEMENTARY & SECONDARY TEACHERS
ELEMENTARY AND SECONDARY TEACHERS DEVELOP AND PLAN TEACHING
MATERIALS AND PROVIDE CLASSROOM INSTRUCTION TO STUDENTS.
ELEMENTARY TEACHERS NORMALLY WORK WITH ONE GROUP OF PUPILS
DURING THE ENTIRE SCHOOL DAY.  SECONDARY TEACHERS USUALLY
SPECIALIZE IN THEIR TEACHING SUBJECTS.  SPECIALTIES INCLUDE
EDUCATING THE HANDICAPPED, COACHING SPORTS, AND TEACHING VO-
          :                :                  :
```

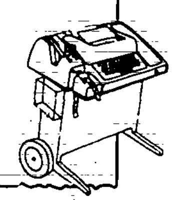


Figure 2-33. Sample interaction with CIS.

```
?PREP 8400
 8400   SOCIAL SERVICE OCCUPATIONS:
PERSONS IN SOCIAL SERVICE OCCUPATIONS PROVIDE DIRECT
SERVICES TO PEOPLE.  THEIR PRIMARY FUNCTION IS TO EN-
HANCE THE PERSONAL DEVELOPMENT OF INDIVIDUALS THROUGH
TEACHING, COUNSELING, ETC.  CAREER PREPARATION COULD IN-
CLUDE THE FOLLOWING HIGH SCHOOL SUBJECTS:
    ARITHMETIC      COMPOSITION         READING
    BIOLOGY         POLITICAL SCIENCE   SOCIOLOGY
    CHILD CARE      PSYCHOLOGY          SPEECH
SOME HIGH SCHOOLS OFFER SPECIAL COURSES IN THE PUBLIC
SERVICE CLUSTER.  ON THE OTHER HAND, STUDENTS MAY NEED TO
CONSIDER POST-SECONDARY AND/OR ON-THE-JOB TRAINING.

YOU CAN LEARN MORE BY LOOKING AT THE DESCRIPTION FOR THESE
OCCUPATIONS:

8414 COUNSELORS
8416 CASEWORKERS
8418 PSYCHOLOGISTS
     :                      :                    :

?VISIT 8456
 8456   ELEMENTARY & SECONDARY TEACHERS
PERSON TO CONTACT:            FIRM NAME:
JAMES E. SHAW                 MONROE JR. H. SCHOOL
SECONDARY TEACHER             2800 BAILEY LANE
PHONE: 342-5611               EUGENE
SPECIAL INSTRUCTIONS: CALL EXT. 254 AND ARRANGE TIME WITH SECRETARY
WILL ALSO TALK TO GROUPS IN EUGENE/SPRINGFIELD AREA

YOU MAY CONTACT ANY PERSON ON THE VISIT LIST.  EACH
ONE HAS VOLUNTEERED TO DISCUSS HIS OR HER OCCUPATION.
(THIS IS NOT A REFERRAL TO A JOB INTERVIEW).

?PROG 236
236   SECONDARY EDUCATION

     MANY 4-YEAR COLLEGES OFFER PROGRAMS INTENDING TO
PREPARE SECONDARY SCHOOL TEACHERS.  SOME SCHOOLS ALSO
OFFER SPECIFIC COURSES FOR STUDENTS WANTING TO SPECIALIZE
IN JUNIOR HIGH SCHOOL TEACHING.  IN ADDITION, SEVERAL COMMUN-
ITY COLLEGES OFFER ASSOCIATE DEGREE PROGRAMS INTENDING TO
QUALIFY PEOPLE IN BUSINESS & INDUSTRY TO TEACH VOCATIONAL
COURSES.  COMPLETION OF AN APPROVED SECONDARY EDUCATION
BACHELOR'S DEGREE IS A PRE-REQUISITE FOR EARNING A BASIC
TEACHING CERTIFICATE.  TO TEACH BEYOND THREE YEARS REQUIRES
     :                      :                    :
```

Figure 2-33. continued.

An example is shown in Figure 2–34 of an interaction with a worksheet generator program. Such programs may allow the user to specify a limited number of characteristics for the instructional materials, as in the example in Figure 2–34, where the number and kinds of problems alone are specified; or they may allow a much broader range of specifications in order to produce truly tailor-made worksheets for class or individualized instruction.

Test construction is another area in which the computer has proved itself a versatile and efficient materials generator. Using an appropriate test-construction program, a teacher can readily obtain test printouts to use as final examinations, exercises, achievement tests, or diagnostic aids.

There are two basic types of test generator programs. Most general-purpose test-construction programs use an appropriate "item bank" from which prewritten test items are selected according to the criteria specified by the teacher—subject matter, classifications within subject matter, key words, behavioral category (e.g. knowledge, comprehension, application, analysis, synthesis, and evaluation), behavioral objectives, statistical characteristics, or item difficulty. Using such a program, tailor-made tests can be easily produced for class use or individual testing.

Tests involving unique test items can also be generated. The programs require the teacher to specify a model of a test item's structure and the rules for altering it; for example, a teacher might specify such a simple item model as:

> On a vacation trip of *A* miles, Sue drove *B* miles the first day. What percent of the total mileage did she drive on the first day?

The rules for altering this model might be:

generate values for *A* and *B* such that:
A is greater than *B* + 50;
A is less than 4,000 miles;
B is greater than 50;
B is less than 600.

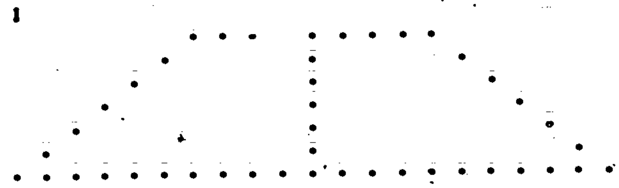
Once the model and rules have been entered, the typical unique-item construction program will then produce a number of unique items, all having the same format but with different variables and different correct answers. Thus a teacher can design a single model with rules from which the computer can generate one-of-a-kind sets of exercises or a uniquely different exam for each individual student and, at the same time, the teacher's answer key for each test.

In addition to the above instructional materials, computer programs are being used to produce other materials potentially useful for instruction—or recreation: "Computer Art," pictures of Snoopy or a pinup girl, calendars, dot-to-dot games, maps, or mazes. Figure 2–35 shows a sample run of the maze-generating program AMAZIN from the University of Oregon's PDP–10 BASIC library. Can you think of a way such a program could be used to enhance instruction?

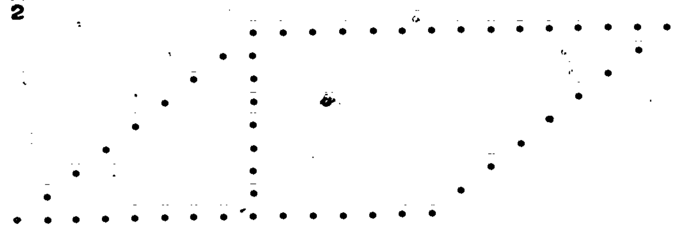
Computer-Managed Instruction (CMI).

The instructional roles the computer can play in the classroom ordinarily rely on the teacher to keep track of student progress and to guide, test, and instruct between and around appropriate student-computer interactions. The computer itself, however, can be of additional help to the teacher by taking over some of these tasks; in fact, whenever individualized instruction is a serious

```
>ØLD AIR*
>NHR

THIS PRØGRAM GENERATES WØRKSHEETS ØN PERIMETER
AND AREA ØF PARALLELØGRAMS AND TRAPEZØIDS.
HØW MANY WØRKSHEETS DØ YØU WANT?  (35 IS MAX.)  ? 1
HØW MANY PRØBLEMS ØN EACH WØRKSHEET?  (10 IS MAX)  ? 2
YØU HAVE THE FØLLØWING CHØICES FØR UNITS ØF MEASUREMENT:
1.  INCHES
2.  CENTIMETERS
3.  NØNSTANDARD UNITS
WHAT'S YØUR PLEASURE? 2
YØU HAVE THE FØLLØWING CHØICES FØR THE TYPE ØF PRØBLEMS:
1.  ALL RECTANGLES
2.  PARALLELØGRAMS THAT ARE NØT RECTANGLES
3.  ALL TRAPEZØIDS
4.  A RANDØM SELECTIØN ØF THE ABØVE CHØICES
WHAT'S YØUR PLEASURE? 4

WØRKSHEET # 1  FIND THE PERIMETER AND AREA ØF EACH FIGURE.
USE A RULER MARKED IN CENTIMETERS TØ HELP YØU.

1
```



```
2
```



```
-------------------------
ANSWERS.  WØRKSHEET # 1
THE UNIT ØF DISTANCE = THE CENTIMETER.
 1 PERIMETER= 22.159213 AREA= 18.064480
 2 PERIMETER= 24.804285 AREA= 24.085973
-------------

 TIME :     .657 / 1/0 : 12
```

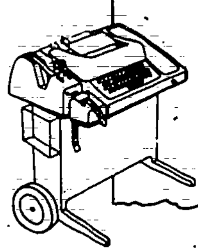


Figure 2-34. Sample interaction with worksheet generator, AIR, available from Minnesota Educational Computing Consortium.

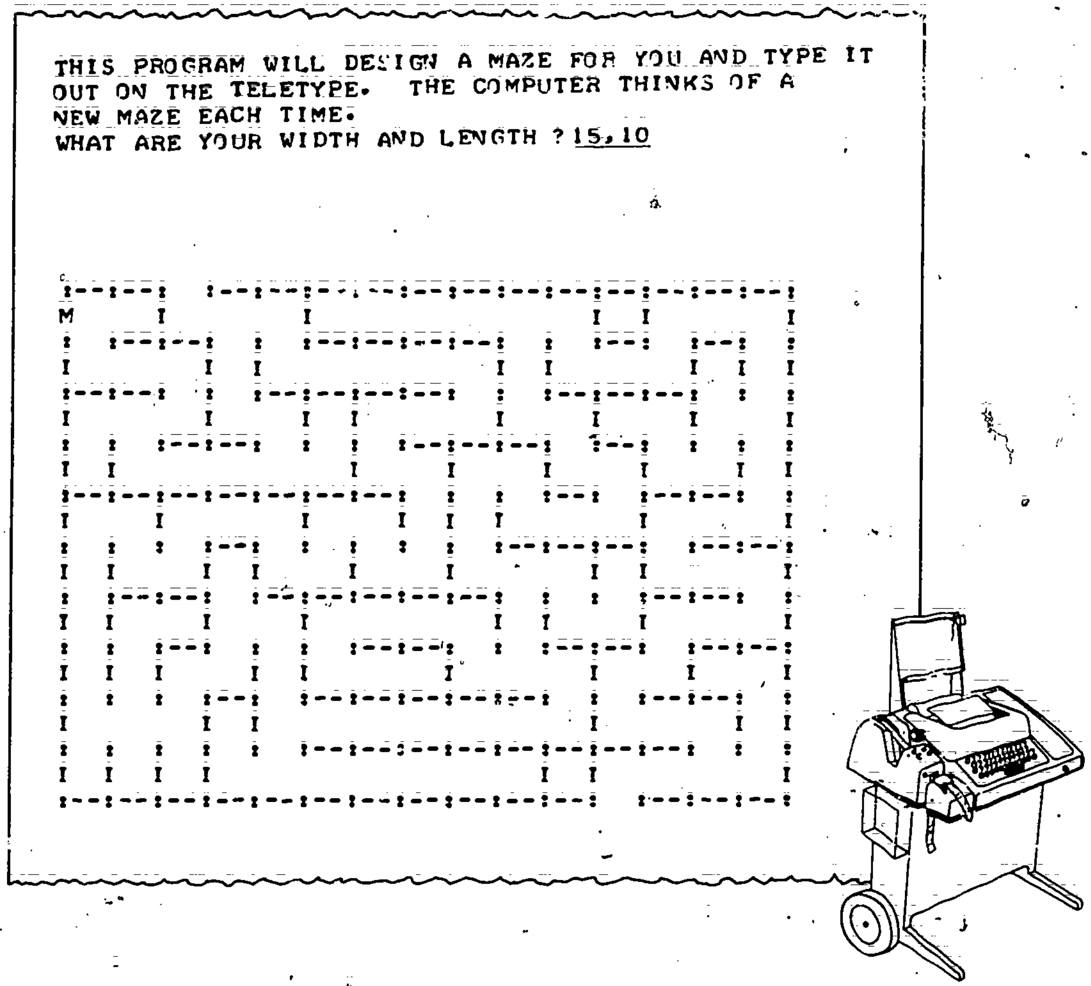


Figure 2-35. Sample interaction with AMAZIN. Available from Minnesota Educational Computing Consortium.

goal, a Computer-Managed Instruction (CMI) system is virtually required. Such systems are designed specifically to help teachers with the considerable number of record-keeping and clerical tasks involved when instruction is individualized.

A variety of CMI systems are currently available. All of them typically perform the basic tasks of storing student records and profiles and using the information to help analyze student progress and direct the learning sequence. On the basis of test scores entered into the CMI system, for example, the student's performance may be evaluated against previous data and directions for further work may be given—either remedial, advanced, or corollary. Thus some of the routine kinds of work are taken over by the computer, and appropriate information is made available for teacher decision-making.

Existing CMI systems range from very simple to very comprehensive, and some incorporate other modes of computer use. At John Abbott College in Quebec, for example, a physical science

CMI system called Computer Generated Individualized Problem Sets generates completely individualized problem sets from teacher-supplied algorithms and prototypes, constructs other tests from an item bank, maintains records of students' progress, automatically scores tests and provides diagnosis of student errors, and performs an item response analysis. Although this system is limited to physical science courses, it performs a wide range of extremely helpful functions within that area.

Another system, the Cincinnati Instructional Management System, has been developed as a teacher's assistant in reading and mathematics. Here the computer does not generate tests or problem sets, but does score the pencil-paper tests used to diagnose and track student progress. It also keeps student records and produces diagnostic reports by student, group, and class. At the request of the teacher, it will make review or study assignments.

These two systems demonstrate the variety of functions a CMI system might carry out. Usually, they score and/or analyze tests, diagnose or identify instructional needs, and prescribe or select appropriate educational experience. All of this information is supplied to the teacher to assist in far more inclusive instructional decision-making than is normally possible. In addition, CMI systems often are equiped to generate tests or instructional resource lists, to provide achievement profiles for individuals or groups, to group students by ability, needs, and progress, and to perform a range of statistical analyses, such as test item analysis.

A CMI system can probably best be thought of as an umbrella for individualized instruction, under which lies a range of instructional and evaluation experiences. For example, a teacher may select a given mathematics objective from a computer-linked "library" of objectives for the students to master; a CMI system, meanwhile, could generate and score diagnostic tests and prepare an achievement profile for each student. Using these profiles, the system could produce an individualized prescription for each student—perhaps a textbook reading assignment, 15 minutes of one-to-one tutoring by the teacher, half an hour of computer-based drill and practice, and completion of a computer-generated practice problem set. Finally, the computer might generate a criterion-referenced test to measure the student's achievement of the objective. If it were adequate, the student would go on to the next objective; if not, the computer would generate a new prescription for review or further study based on the test items on which the student performed poorly.

As you can see, CMI has the capability of providing the modern teacher with a vigorous support system for accomplishing a wide range of instructional goals from simple, coherent tracking procedures to the most inclusive individualized instruction aims.

III

SELECTING COMPUTER-BASED INSTRUCTIONAL UNITS

SELECTING COMPUTER-BASED MATERIALS: SOME BASICS

Introduction

> Quality . . . you know what it is, yet you don't know what it is. But that's self-contradictory. But some things *are* better than others, that is, they have more quality. But when you try to say what the quality is, apart from the things that have it, it all goes poof! There's nothing to talk about. But if you can't say what quality is, how do you know what it is, or how do you know that it even exists? If no one knows what it is, then for all practical purposes it doesn't exist at all. But for all practical purposes it really *does* exist. What else are the grades based on? Why else would people pay fortunes for some things and throw others in the trash pile? Obviously some things are better than others . . . . But what's the "betterness"? . . . So round and round you go, spinning mental wheels and nowhere finding anyplace to get traction. What . . . is quality? What *is* it?
>
> (Robert M. Pirsig, *Zen and the Art of Motorcycle Maintenance*)

Judging quality is difficult. Even so, we are continually making quality judgments—about our students and about the instructional materials available to them.

Often we make intuitive or gut-feeling (subjective) judgments and they turn out to be quite reliable; other times we apply objective, explicit standards. Both approaches involve the application of some criteria against which we are comparing the thing to be judged. The criteria may be unconscious, forming the basis for strong intuition, or they may be clearly stated, even systematized.

In an effort to help educators make objective judgments about instructional products, many systems or models have been developed. Generally, these ask the evaluator to make quality judgments on a number of different characteristics of the product.

Most of the models currently available for judging or evaluating materials would also be useful in evaluating computer-based instructional products, but they would need to be expanded to include careful examination of some characteristics specific to the computer.

In the following pages we present a simple yet usable set of guidelines for evaluating computer-based instructional materials. It is not intended to be exhaustive; it presents only the basic considerations showing one way to evaluate and select computer-based instructor units for possible classroom use.

In this section, we will focus specifically on the instructional applications which fall into the first five categories of computer use discussed in Part II, including drill-and-practice, tutorial, data analysis, problem solving and simulation:

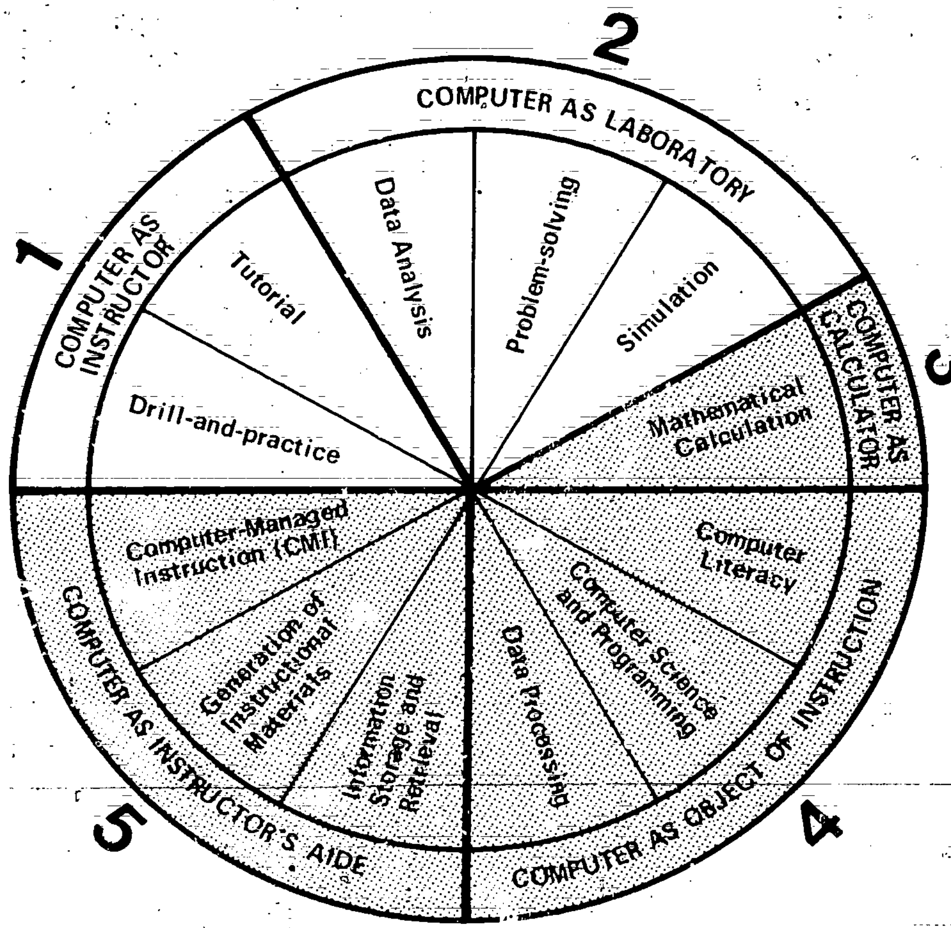


Figure 3-1 Instructional units in the first five categories.

Variety of Computer-Related Materials

A wide variety of computer-related instructional applications is currently available from different sources, including university projects, hardware manufacturers, and educational corporations; a list is given in Appendix II. As an indication of the volume of available applications, a study at HUMRRO in 1975 identified 5,650 items of computer-based curricular materials. The 1976 edition of the *Index to Computer Based Learning* documents and abstracts 1,837 items.

There is a wide variation in completeness, orientation, and usability of the units currently available. Comparing these units for selection for classroom use may be complicated by the fact that the support materials provided may range from a single program listing to a complete volume of workbooks, manuals, and program guides. In addition, there are the computer-related considerations to add to your regular set of criteria for judging the usefulness of materials for your classes. For these reasons, selecting computer-based instructional units can be a complicated and confusing job. The procedure outlined here, however, is designed to make the process logical and easy, since it concentrates on the basic components and discusses the general characteristics which contribute to their usability.

A Computer-Based Instructional Experience

Before beginning to select computer-based applications, you should have a clear idea of what elements are basic to a computer-enhanced instructional experience. The components are illustrated in Figure 3-2.

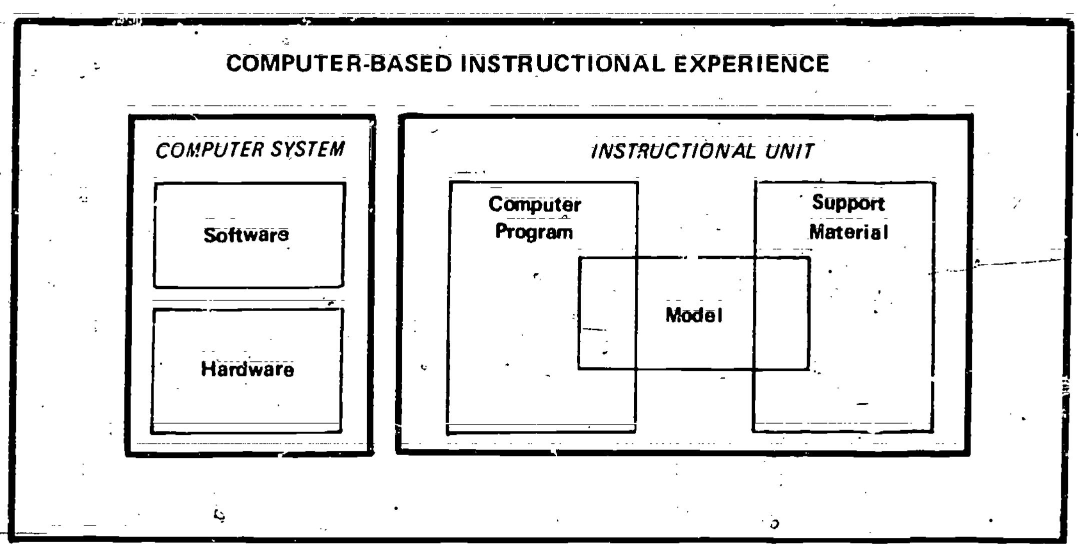


Figure 3-2 Elements of a computer-based instructional experience.

As you see, there are two main parts—the computer system and the instructional unit.

The first component is the computer system—hardware and software. As discussed in detail in Chapter I, the appropriate computer hardware (equipment) and software (program and language capabilities) must be available to enable the unit to be used effectively or at all.

The instructional unit includes a computer program, a model or algorithm around which the unit is organized, and, in some cases, student and teacher support materials. Very often, particularly if the unit was developed by a busy teacher, the unit will include minimal, if any, support material. In such cases, the teachers who plan : the unit may have to expand or even develop their own materials for classroom use.

In the diagram of the instructional unit (Figure 3-2) note that the model is shown to overlap with the program and support materials; this is intended to illustrate that the model is formulated independently but is contained in the program and is described and explained in the support materials. When considering a computer-based unit, all of these components of the computer-based instructional experience must be evaluated—the hardware and software requirement and the program, model, and support materials.

The Process of Selection: Two Stages

The process of selecting computer-based units can be divided into an initial stage, in which decisions are usually dictated by basic, practical requirements, and a final stage, in which evaluations and comparisons are more often based on personal likes and dislikes or on accepted

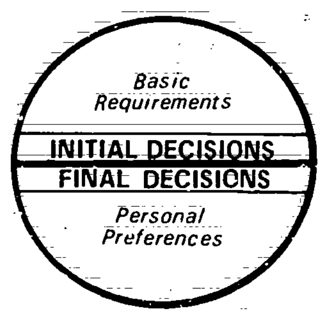


Figure 3-3. Decision stages—initial and final.

standards. The first stage is concerned with the practical necessities of whether or not a unit can be used with the available equipment and whether or not it fulfills the teacher's basic requirements: the teacher determines whether the unit is fundamentally usable and therefore worth evaluating further. The second stage evaluates the more detailed aspects of the unit, and decisions are determined by teacher preferences and personal or established standards.

BEGINNING THE SELECTION PROCESS

The Initial Decision

When you begin to select from among computer-based units, the first stage of your evaluation will naturally focus on determining whether or not the unit is usable on the computer you have available, teaches what you want to teach, and has at least the minimum materials you need to incorporate the unit into your class work. These basic considerations can be expressed as five critical questions concerning the computer system and the instructional unit:

1. Will the program run on the hardware I have available? If not, can it be altered?

2. Is the language available on my computer? If not, can the program be rewritten in an available language without destroying the instructional value?
3. Does the unit encompass appropriate instructional objectives, instructional strategy, and difficulty level for my purposes and my students?
4. Is the model used in the unit accurate and appropriate? If not, can it be revised to my specifications?
5. Are the support materials useful and complete enough to meet my minimum standards? If not, can I either provide the necessary revisions or develop support materials to my specification?

These preliminary questions can be called *absolute* decision variables, since a No answer to any question will usually eliminate the unit from further consideration.

The preliminary set of decision variables can be illustrated in a table, as in Figure 3–4, where units are evaluated on the five absolute decision points as NO, YES, or MAYBE (i.e. usable with modifications).

INITIAL DECISION VARIABLES	POTENTIALLY USABLE UNITS				
	Unit #1	Unit #2	Unit #3	Unit #4	Unit #5
1. Will program run on hardware?	NO	YES	YES	YES	YES
2. Is language available?		YES	MAYBE	YES	NO
3. Appropriate objective, strategy and difficulty level?		MAYBE	MAYBE	MAYBE	
4. Is model accurate and appropriate?		YES	YES	NO	
5. Are support materials sufficient?		YES	NO		
PRELIMINARY DECISION	REJECT	EVALUATE FURTHER	REJECT	REJECT	REJECT

Figure 3–4. Sample preliminary decision table.

By the time you have asked and answered all the preliminary questions, you may well have rejected the unit. But if the answers are all Yes or Maybe (as with Unit 2 in the example), the final decision will depend on how you rate the unit on the secondary characteristics in the second stage of evaluation, as we will see.

1. Will Program Run on Hardware?

It may seem biased (even heretical!) to focus on the question of hardware before we have considered instructional objectives. It is reasonable to do so, though, because the answer may make the final decision at the outset. There is no point in making an exhaustive investigation into the instructional objectives of the unit only to discover later that the program requires a computer with twice the capacity of your own and must run in interactive mode while you have only batch mode available.

A math teacher or student, or someone else in your school familiar with the local computer facilities, can probably give you a quick answer to this question simply by looking at the program listings for any units you want to consider. Or, the support materials may provide information about the size computer and type of terminal required to run the unit's program* so that you can judge from your own knowledge whether the program will run. This variable should involve only a Yes or No decision.

2. Is Language Available?

The next most critical variable is the programming language used; and again, this can be answered quickly by someone familiar with your computer system. By studying Appendix II, Keys to Reorganizing General Purpose Languages, you can probably determine for yourself what language the program is written in and then find out whether that language is available.

Even if the language is available on your computer, modifications may need to be made, requiring anywhere from five minutes to several days of an experienced programmer's time. Modifications are commonly required because different computers use slightly different versions of the same language, but in most cases they are minor. Again, you should consult with someone familiar with your local computer.

If the language is not available, a larger programming task is required to convert the program to a language that is available. If you have no programming skills (or no time) and no access to a programmer, this probably means you will have to eliminate this application.

3. Appropriate Objectives, Strategy, and Difficulty Level?

Once the practical hardware/software questions have been examined, you can begin considering the instructional validity of the unit.

OBJECTIVES

Assuming the support materials include a statement of the instructional objectives (and that you have defined your own objectives), you must determine whether the stated objectives are consistent with your own. If they are not, you should be able to eliminate the obviously inappropriate units fairly easily. If you want to teach eighth-grade students to recognize and define Greek word roots for a unit on English vocabulary, for example, you would obviously eliminate a program designed to teach twelfth-grade students Greek grammar or a program aimed at teaching the development of Greek culture.

The objectives of many units may be near enough to yours, or general enough, that some modification in the unit materials, or even in the objectives themselves, could tailor them to your purposes.

INSTRUCTIONAL STRATEGY

The five uses of computers in the first two categories of computer use, discussed on pages 34–60, reflect a variety of teaching modes. You may have a clear preference for one or another of these for certain topics or certain students. One dimension of the student-computer interaction you may wish to consider is the degree of learner control *vs.* computer control. The five types of applications can be arranged roughly along a control continuum as shown in Figure 3–5.

Important differences in teaching philosophy are demonstrated here. With tutorial and drill-and-practice programs, the student is asked only to respond to the information presented by the

*See Part I, pages 12–14.

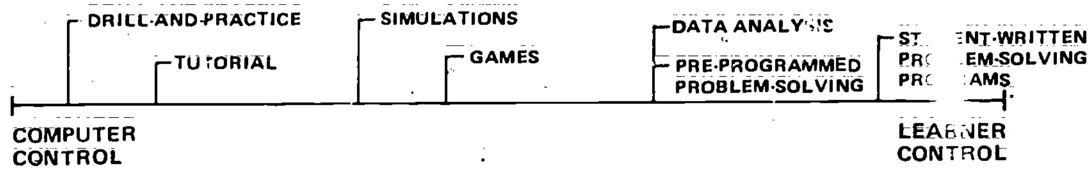


Figure 3-5. Control continuum for computer uses in instructional applications.

computer, which then judges the response and presents new information. Games and simulations add a dimension by manipulating the data input by the student according to an intrinsic model (set of rules) and displaying an outcome that is the direct result of the student's action; thus the student learns actively to exercise the model in order to control the outcome. Farther along, when the student uses the computer as a tool to calculate, analyze data, or solve a problem, he controls the tool by providing data to get a needed answer. Finally, when he actually writes the program, he is controlling the computer by providing not only the data but the program itself.

DIFFICULTY LEVEL

Appropriate "difficulty level" is another way of saying that your students will have all the prerequisite skills and knowledge required to interact with and learn from this particular program—rudimentary typing skills and appropriate vocabulary level, for example, as well as knowledge. Students also should be capable of the level of behavior the program requires with respect to knowledge, comprehension, application, analysis, synthesis, or evaluation. Probably the best way to make a preliminary judgment about the difficulty of the program is to run the program or to examine a sample run. As an example, suppose you wish to find a unit on perimeters and areas of quadrilaterals for your seventh grade mathematics class. You locate a computer program which is identified as dealing with quadrilaterals in geometry. Upon examining a sample run, however, you find that it was obviously developed for use with high school students who have had experience with "proofs" and your students would not be able to use it. You must either find a way for the students to acquire the necessary prerequisite skills and knowledge, revise the program, or reject it. You may decide against other programs because they are too simple or use condescendingly elementary language or tone. In most cases, these considerations will be less than absolute. It is possible, for example, that a unit designed for ninth graders could be used with tenth, eleventh, or twelfth graders, or that a particular business education unit might be used just as effectively in a social studies class. The final decision must depend on the match between the unit's objectives and your own.

4. Is the Model Appropriate?

Three of the instructional modes we are considering—simulation, problem-solving, and data analysis—are built around models. As part of your initial decision to accept or reject an instructional unit, it is important therefore to determine the appropriateness of the model underlying a particular unit.

In the case of data analysis the underlying model is mathematical or, more often, statistical in nature. In other words, data analysis programs involve using a data analysis model to analyze and interpret the data. When you have data you wish to evaluate, these models help organize them in such a way that relationships existing within them can be discovered. Suppose, for example, that you have conducted a sample survey to determine students' attitudes toward school.

The data you collect can be evaluated with statistical models to help you determine whether or not the sample is biased, what attitudes are related to other attitudes, how strong these relationships are, and so forth. You are probably familiar with several data analysis models. Correlation is one; the chi square test and t-tests are others. Most statistical models included in computer statistical packages are well accepted within the research community and you can accept them as valid. In addition, most elementary statistics texts contain chapters devoted to the various statistical models.

The role played by models in simulations can be depicted in the diagram in Figure 3-6.

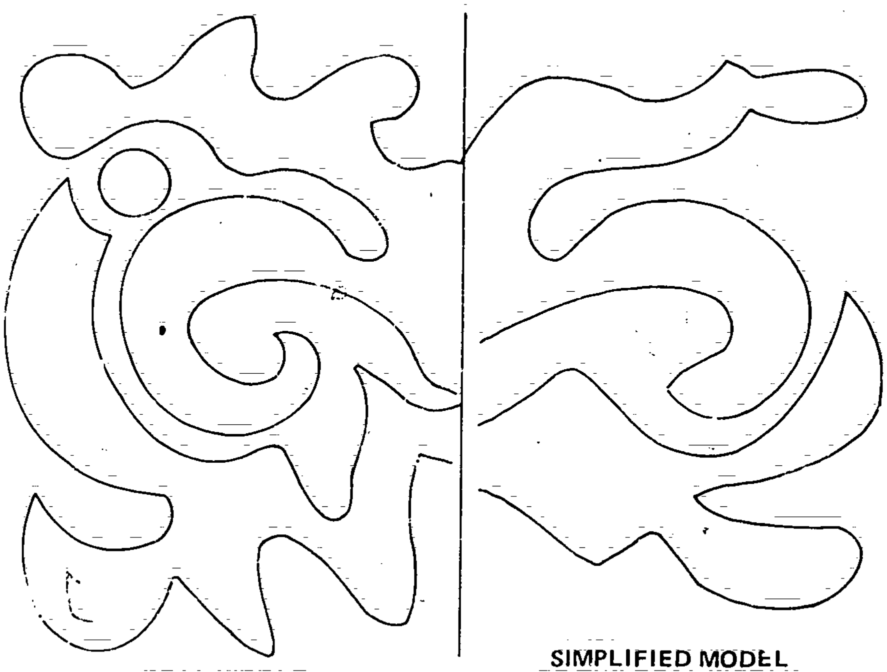


Figure 3-6. The model reflects the real world in a simplified form.

Figure 3-6 suggests that any real world situation can be represented by a model. The real estate game called *Monopoly*; a Link Trainer for airplane pilot training; a map of Minneapolis, Minnesota; and a Mickey Mouse cartoon film are all models of one type or another.

Simulations are operating models of physical or social situations. The word model in the context of simulation implies two things—first, that the reality being represented is reduced in size to manageable proportions; and second, that only certain aspects of the real thing are chosen for inclusion. In other words, reality is simplified.

Problem-solving models are usually quite specific. Examples would be the formula for converting temperatures from Fahrenheit to Celsius, or a formula to project population growth. Any model in a computer-based instructional unit is stated in mathematical terms within the computer program. Most teachers will find it difficult to determine what model is being used by looking at a program, but it should be described in simple terms in support material. Pages 15-20 of the

URBAN CRIME UNIT provide a good example of a verbally stated model. The more mathematical statement appears on pages 34–35.*

For another example, look at the social studies unit called "Balance of Trade and Balance of Payments," reprinted here in its entirety.† The support materials consist of a one-page cover-sheet, a sample run of the program, and a program listing. From an examination of these materials, see if you can determine what model was used.

You no doubt discovered that the sample run gives the formulas used to computer trade balance and balance of payments:

```
THADE BALANCE = EXPORTS-IMPORTS.

BALANCE OF PAYMENTS = ALL OVERSEAS EXCHANGES+
ALL OVERSEAS EXPENDITURES.
```

This is part of the model. From a further examination, you can see how the figures circled below were derived:

```
A. FOREIGN AID =              25
B. BALANCE OF TRADE =        233
C. TRAVEL BALANCE =          198
D. INVESTMENT BALANCE =       2:
----------
   BALANCE OF PAYMENT         6:
```

The total for Foreign Aid is the sum of the two input values 12 and 13. Balance of Trade was simply computed according to the formula, using the two input values 256 and 23. Travel Balance is the difference between the two input values 254 and 56. Investment Balance is the difference between the two input values 259 and 21. To verify these observations, you can look at the program listing shown in Figure 3–8.

This, then, is the algorithm or model upon which this unit is based. The model is simple. It is up to you to answer the questions:

1. Is the model accurate enough?
2. In an attempt to highlight critical relationships, has the model been too simple to represent reality accurately?
3. Is the model too complex to allow the student to recognize relationships?
4. Is the model complete enough to meet your objectives?

When using simulations and problem-solving models, it is only good teaching to inform students of the nature of the model they are interacting with—that it is a simplification of reality, that many factors may have been ignored, and that the model determines all output from the computer. Students may wish to investigate the assumptions of the model to determine its validi-

*Available from Digital Equipment Corporation, "Huntington I Applications Programs–Social Studies."
†Available from Digital Equipment Corporation, "Huntington I Applications Programs–Social Studies."

[illegible] STUDIES

SUBJECT: ~~DISTINCTION BETWEEN BAL-~~

~~ANCE OF TRADE AND BALANCE OF~~

PAYMENTS

PROGRAM NAME BALANC

DESCRIPTION:

This program demonstrates the distinction between "balance of trade" and "balance of payments." Also shown are the components that make up the "balance of payments" account, and their individual impacts.

OBJECTIVES:

A. To emphasize the important distinction between "Balance of Trade," and "Balance of Payments."

B. To demonstrate the impact of any specific foreign expenditure on our "Balance of Payments."

PRELIMINARY PREPARATION:

A. Student must obtain data for components of balance of payments for a given year and country.

B. Discussion of the concepts "balance of trade" and "balance of payments", would be helpful but are not necessary.

DISCUSSION:

A. Student level - average

B. Curriculum location - advanced economics: Unit on U. S. Economy in the World.

C. This program may be used either as a group exercise, or for individual study.

1

```
THERE'S A DISTINCTION BETWEEN TRADE BALANCE AND
BALANCE OF PAYMENTS.

TRADE BALANCE = EXPORTS-IMPORTS.

BALANCE OF PAYMENTS= ALL OVERSEAS EXCHANGES+
ALL OVERSEAS EXPENDITURES.

WHEN INFORMATION IS REQUESTED, INPUT VALUES IN
MILLIONS OF DOLLARS (E.G. 6 = 6 MILLION DOLLARS)

A.
INPUT A FIGURE FIRST FOR ALL MILITARY AID, THEN
FOR ALL OTHER AID TO OTHER NATIONS.
? 12,13

B.
INPUT A FIGURE FIRST FOR EXPORTS, THEN FOR IMPORT
? 256,23

C.
INPUT A FIGURE FIRST FOR FOREIGNERS TRAVELING IN
YOUR COUNTRY, THEN FOR YOUR COUNTRYMEN TRAVELING
ABROAD.
? 234,36

D.
INPUT A FIGURE FIRST FOR INCOME FROM FOREIGN INVEST-
MENTS, THEN FOR FOREIGN INVESTMENT ITSELF.
? 259,21

A. FOREIGN AID =            25

B. BALANCE OF TRADE =       233

C. TRAVEL BALANCE =         198

D. INVESTMENT BALANCE =     238

-----------             ---------

BALANCE OF PAYMENTS =       644

(REMEMBER, IF A MINUS FIGURE APPEARS ABOVE,YOUR
COUNTRY HAS A DEFICIT IN ITS BALANCE OF PAYMENTS)

HOPE YOU UNDERSTAND THE DISTINCTION BETWEEN THE
BALANCE OF TRADE AND THE BALANCE OF PAYMENTS
BETTER NOW.

READY
```

2

Figure 3-7. Cover sheet and sample run of program BALANC from "Balance of Trade and Balance of Payments" unit.

```
1 REMPROGRAM BY J.V.SWARTZ, HALF HOLLOW HILLS,7/68
5 PRINT"THERE'S A DISTINCTION BETWEEN TRADE BALANCE AND"
7 PRINT"BALANCE OF PAYMENTS."
8 PRINT
10 PRINT"TRADE BALANCE = EXPORTS-IMPORTS."
15 PRINT
20 PRINT"BALANCE OF PAYMENTS = ALL OVERSEAS EXCHAN
22 PRINT"ALL OVERSEAS EXPENDITURES."
23 PRINT
25 PRINT "WHEN INFORMATION IS REQUESTED, INPUT         IN"
27 PRINT "MILLIONS OF DOLLARS (E.G. 6 = 6 MILL      LARS)"
30 PRINT
33 PRINT "A."
35 PRINT"INPUT A FIGURE FIRST FOR ALL MILITARY AID, THEN"
36 PRINT"FOR AL      ER AID TO OTHER NATIONS."
45 INPUT F1,
50 LETF=F1+F2
52 PRINT
54 PRINT "B."
55 PRINT"INP      RE FIRST FOR EXPORTS, THEN FOR IMPORTS."
65 INPUT S1,S2
70 LETS=S1-S2
72 PRINT
74 PRINT "C."
75 PRINT"INPUT A FIGURE FIRST FOR FOREIGNERS TRAVELING IN"
76 PRINT"YOUR COUNTRY, THEN FOR YOUR COUNTRYMEN TRAVELING"
77 PRINT"ABROAD."
85 INPUTT1,T2
90 LET T=T1-T2
92 PRINT
94 PRINT "D."
95 PRINT"INPUT A FIGURE FIRST FOR INCOME FROM FOREIGN INVEST-"
96 PRINT"MENTS, THEN FOR FOREIGN INVESTMENT ITSELF."
105 INPUT I1,I2
110 LET I=I1-I2
115 LETP=-F+S+T+I
117 PRINT
118 PRINT
120 PRINT "A. FOREIGN AID =",F
125 PRINT
130 PRINT "B. BALANCE OF TRADE =",S
140 PRINT
145 PRINT "C. TRAVEL BALANCE =",T
150 PRINT
155 PRINT "D. INVESTMENT BALANCE =",I
158 PRINT
160 PRINT "-----------"," ","-----------"
162 PRINT
170 PRINT "BALANCE OF PAYMENTS =",P
171 PRINT
172 PRINT"(REMEMBER, IF A MINUS FIGURE APPEARS ABOVE,YOUR"
173 PRINT"COUNTRY HAS A DEFICIT IN ITS BALANCE OF PAYMENTS)"
181 PRINT
185 PRINT"HOPE YOU UNDERSTAND THE DISTINCTION BETWEEN THE"
186 PRINT"BALANCE OF TRADE AND THE BALANCE OF PAYMENTS"
187 PRINT"BETTER NOW."
190 END
```

Computes foreign aid (F) by computing the sum of all military aid (F1) + aid to other nations (F2).

Computes balance of trade (S) by subtracting imports (S2) from exports (S1).

Computes travel balance (T) by subtracting countrymen traveling abroad (T2) from foreigners traveling in your country (T1).

Computes investment balance (I) by subtracting foreign investments (I2) from income from foreign investment (I1).

Computes balance of payments (P) according to formula: To find balance of payments (P), subtract foreign aid (F) from the sum of balance of trade (S) + travel balance (T) + investment balance (I). So:
P = -F + S + T + I.

Figure 3-8. Program listing for BALANC with explanatory comments.

ty for themselves. In fact, a good deal of deductive learning takes place when students are allowed to build their own models or improve upon models developed by others.

On the following pages is another example of how a model can be described. It is taken from the Huntington II Simulation Program POLUT.*

Is the description of the POLUT model informative enough to determine the appropriateness of the model?

*Available from Digital Equipment Corporation.

V. THE POLUT MODEL

The defining equations and the constants used in the model are as follows:

$$\frac{dW1}{dt} = D1 - NW1 \qquad (1)$$

$$W1(0) = D1/N \qquad (2)$$

$$\frac{dW2}{dt} = D2 - HW2 \qquad (3)$$

$$\frac{dX}{dt} = C(X9 - X) - NW1 - HW2 \qquad (4)$$

$$W = W1 + W2 \qquad (5)$$

$$X(0) = X9 - D1/C \qquad (6)$$

$$N = 0.75 \qquad (7)$$

$$H = 0.25, \text{ for industrial waste, or } 0.75 \text{ for sewage} \qquad (8)$$

$$W2(0) = 0 \qquad (9)$$

$$D1 = 2 \qquad (10)$$

$$C = \begin{cases} 0.4 & \text{for pond} \\ 1.0, & \text{lake} \\ 1.5, & \text{slow river} \\ 3.0, & \text{fast river} \end{cases} \qquad (11)$$

in which

W1 = waste due to natural pollutants (dead fish, leaves, etc.) in parts per million (ppm).

W2 = waste due to humans (in ppm).

D1 = rate of injection of natural pollutants (in ppm per day).

D2 = rate of injection of human pollutants (in ppm per day).

N = waste decomposition coefficient for natural wastes

H = waste decomposition coefficient for human wastes (sewage or industrial).

X = dissolved oxygen level (in ppm).

X9 = saturation oxygen level (function of temperature) (in ppm).

W = total waste in water (in ppm).

C = rate of absorption of oxygen in water (in ppm per day).

Equation 1 states that the change in the amount of natural pollutants in the water from day to day equals the rate at which natural pollutants are injected minus the rate at which these pollutants are decomposed. Notice that the waste due to natural pollutants at any time is constant, unless the rate of injection of natural pollutants changes with time [i.e., W1 = W1(0) for all t>0]. For this program, the assumption has been made that the rate of generation of natural wastes is constant throughout the year, but Equation 1 is retained in the event that a more elaborate model is desired at some future time.

Equation 2 ensures that the water system is in equilibrium at day 0 (i.e., before human wastes begin polluting the water).

According to Equation 3, the change in the amount of human pollutants in the water from day to day equals the rate at which human pollutants are injected minus the rate at which these pollutants are decomposed. HW2 represents the rate of decomposition of human waste. Note that when the human waste is sewage, the decomposition coefficient, H, takes on the same value as the decomposition coefficient for natural wastes. The value of H is considerably higher for sewage than for industrial waste.

Equation 4 states that the dissolved oxygen level of a body of water changes from day to day, depending on the difference between the saturation level and the actual oxygen level and the rates at which natural and human pollutants are decomposed. It is significant to note here that the saturation level of dissolved oxygen in water decreases as the water temperature increases, according to the formula:

$$X9 = \begin{cases} 15 - 2[(T-32)/9] & \text{for } 32 < T \le 50 \\ 11 - [(T-50)/9] & \text{for } 50 < T < 90 \end{cases}$$

The graph in Fig. 1 depicts this relationship.

Figure 3-9. Description of the POLUT model. (Continued on next page.)

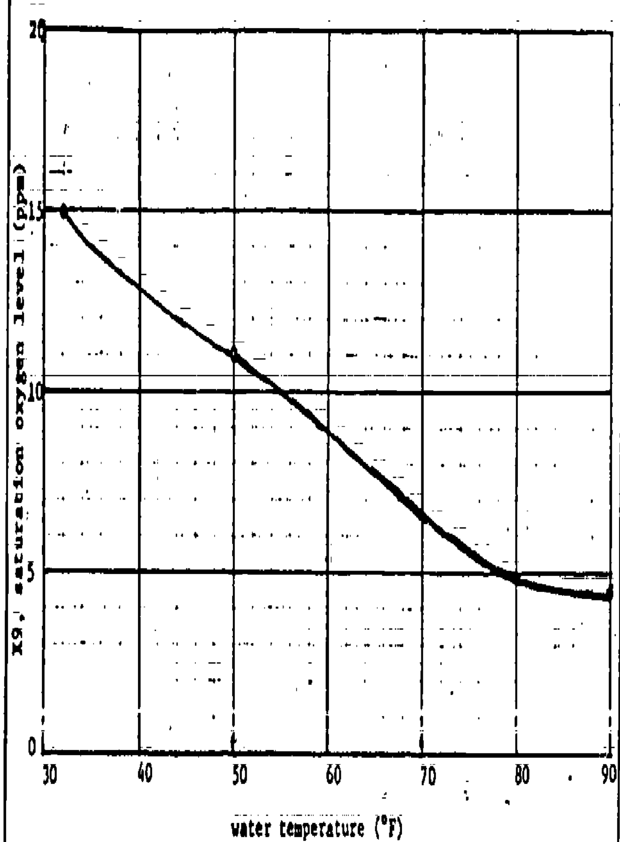


Fig. 1. Water temperature vs. saturation level of dissolved oxygen in water

Equation 5 states that the total waste in the water is the sum of the natural and human wastes in the water.

Equation 6 states that at day 0 (i.e., before human wastes begin polluting the water) the level of dissolved oxygen in the water is less than the saturation level of dissolved oxygen in water by the amount required to decompose the natural waste.

Equations 7-10 state the constant values for waste decomposition coefficients, initial human pollutants, and rate of injection of natural pollutants.

Equation 11 states the rates of absorption of oxygen in water for each type of body of water. In effect, the varying constants reflect the effect of surface area or speed of the body of water on oxygen absorption.

The assumptions under which the model operates are the following:

1. The rate of generation of natural wastes and human wastes is constant throughout the year.
2. Phosphates, nitrogen, dissolved gases, suspended wastes are not important (or, at least, not of interest, and not coupled to our variables).
3. Only natural wastes exist in the body of water prior to the first day on which human pollutants are dumped into the water.
4. There is no interaction between natural and human waste.
5. Natural waste decomposes at the same rate as sewage.
6. The rate of waste decomposition is not dependent on water temperature.
7. The rate of waste decomposition is independent of the oxygen level in the water. As the oxygen level falls, anaerobic organisms begin to take over waste decomposition and maintain the same rate of breakdown.
8. There are no differences in industrial pollutants (e.g., no difference is acknowledged between the wastes from a vegetable cannery and those from a chemical plant).
9. There are no seasonal variations in the water body.
10. Game fish (perch, trout, etc.) die or leave the area when the oxygen level drops below 5ppm.

oo0oo

Figure 3-9. Continued

When examining the model used as a basis for a unit, you will want to make your own sample runs of the program. Once you have inferred the model from the program or descriptive materials, you will evaluate whether it is appropriate for your classes, whether it might be modified, or whether to reject it as inappropriate.

Do not become discouraged if you have difficulty in evaluating a specific model. It is difficult, and there are no simple, easy-to-follow guidelines. Evaluation procedures range from highly complex mathematical measures to gut-level feelings.

5. Are Support Materials Complete Enough?

The minimum standards for the completeness of support materials will vary from teacher to teacher. You may be a teacher who rejects the approved course textbook and instead creates curriculum material from your own resources to meet your standards and objectives. If so, you may prefer to use an instructional unit which includes only a computer program, so that you can develop your own materials and modify the program as you see fit. Or you may find that demands on your time and energies prevent you from developing complete new units but that you have time to modify or expand a unit if certain basic materials exist. Finally, you may be a busy teacher who can use the computer only if good, complete support materials are available.

Some features of support materials to consider in setting minimum standards include:

1. Statement of instructional objectives;
2. Description of the algorithm or model used in the program;
3. Rationale for using the computer as the medium for this instruction task;
4. Prerequisites and preparatory activities;
5. Sequence of classroom activities involved in using the program;
6. Followup activities;
7. Pretest and posttest;
8. Student manual and/or worksheets;
9. Teacher's guide;
10. Resource guide (with background information, text references, etc.);
11. Program listing;
12. Sample run of the program.

You may want to add other features.

As an exercise, use a form like that in Figure 3–10 to list the features of a computer-based unit in order of their importance to you. Then draw a line below your minimum requirements for using the unit in the classroom.

Making the Initial Decision

Once you have determined your own minimum practical requirements and decided whether a unit meets them, you probably will have made a preliminary decision either to reject a given unit or to give it further consideration. If the answers to the five critical questions are all Yes or Maybe, you can base your final decision on the remaining secondary criteria, which we discuss in the next section. If you have answered No to any of the five questions, you probably plan to reject the unit. For example, if you are reviewing eight units and have answered the five questions as in Figure 3–11, you will know what to do next. (Remember, you answer Maybe if you or someone else can revise the unit to make it acceptable.)

WRITE YOUR OWN LIST

1.

2.

3.

4.

5.

6.

7.

8.

9.

10.

11.

12.

13.

14.

15.

Figure 3-10. Minimum requirements list.

In the case of Unit 7, you may decide to reject the unit at this point if you know you don't have the time or resources to modify it to change each Maybe to Yes.

Now you are ready to consider the remaining variables in evaluating a unit and then make your final decision. But for practice, you should first evaluate some sample units in terms of the five initial decision variables, deciding whether or not:

1. The program will run on the hardware you have available in your school;

INITIAL DECISION VARIABLES	POTENTIALLY USABLE UNITS						
	Unit #1	Unit #2	Unit #3	Unit #4	Unit #5	Unit #6	Unit #7
1. Will program run on hardware?	NO	YES	YES	YES	YES	MAYBE	MAYBE
2. Is language available?		YES	MAYBE	YES	NO	YES	MAYBE
3. Appropriate objective, strategy and difficulty level?		MAYBE	MAYBE	MAYBE		YES	MAYBE
4. Is model accurate and appropriate?		YES	YES	NO		MAYBE	MAYBE
5. Are support materials sufficient?		YES	NO			MAYBE	MAYBE
PRELIMINARY DECISION	REJECT	EVALUATE FURTHER	REJECT	REJECT	REJECT	EVALUATE FURTHER	EVALUATE FURTHER

Figure 3–11. Completed preliminary decision table.

2. The language will be available on your system;
3. The objectives, strategy and difficulty level are appropriate;
4. The model is appropriate;
5. The support materials are complete enough.

If you cannot determine, at present, whether or not the hardware or language requirements are appropriate, put Maybe in the column. When you have finished your initial evaluation of the sample units, continue on to the next section.

THE FINAL DECISION

The Final Variables

Once you have selected those units which will run on your hardware and meet your requirements for objectives, materials, and model, you will be ready to evaluate them on the more detailed points which remain:

1. The extent to which the unit enhances, extends, or enriches your own instruction;
2. The extent to which the unit is "student proof"—that is, capable of handling inappropriate input, poor typing, smart-aleck responses, and so forth;
3. The extent to which the input and output are user oriented—that is, simple to read and understand, clearly labeled, concise, and brief;
4. The number of variables that are directly controllable by the user;
5. The number of different options for classroom use;
6. The time and cost involved in using the unit.

Again, you may add variables of importance to you.

As you examine each unit, you will probably find that you cannot make an absolute judgment of the unit's acceptability in each area, but that you will rate the unit for each variable and the rating will give you the basis for your final decisions.

Let's examine some examples which illustrate each of the above variables, with various degrees of acceptability.

1. Extent Unit Enhances Instruction

Sometimes there is a temptation to use the computer as an instructional gimmick—that is, to use computer-based curriculum units simply because "the program works." Indeed, the machine is a fascinating innovation in teaching. It may take some time before it is viewed with the same casual objectivity as a movie projector or tape recorder. In the meantime, teachers must constantly ask themselves, Can I teach this concept or skill as well or better without the computer? Does the computer-based material enhance instruction? As you examine a unit, notice how it was designed to extend the teacher's presentation of the topic by means other than those available in the traditional classroom.

Before rating a unit on its capacity to enhance institution, consider these questions:

1. Which skills or concepts is this unit designed to enhance?
2. What special opportunities for improving these skills are provided by the unit?
3. Could these opportunities be provided by conventional instructional means?

Examine your sample units and obtain your own sample runs. Try to rate each one for its capacity to enhance instruction, using a numeric scale such as:

1 = very poor
2 = poor
3 = good
4 = very good

Put your final rating number in the space provided in the decision table in Figure 3–24 on page 105.

2. Extent to Which Unit is 'Student-Proof'

A program not carefully designed to handle typical student-user problems could well take up undue time for both student and teacher and contribute more to confusion than instruction.

There are three typical areas where students encounter problems using computers. (1) They commonly do not grasp the range of acceptable responses for a program and often enter input which exceeds program limits (too large or too small). (2) Some cannot read scientific notation, which is the way large numbers are commonly output by the computer. (3) In cases of division by zero, incorrect format of input entries, and so forth, the error messages printed out by computers are typically unclear to students.

Programs written for student use should be designed to circumvent these problems. Whenever the student enters input which is outside the acceptable range, the program should print out a message clearly explaining what is wrong and how to correct it. POLUT* is an example of a program designed to take care of the full range of possible input. The student is asked to input the water temperature for the body of water he is studying. The excerpts from the POLUT program run in Figure 3–12 show what happens if the student enters a temperature below freezing or above 90 degrees—outside the allowable input range.

*Program used in Huntington II Simulation Program–POLUT; available from Digital Equipment Corporation.

```
***********

BODY OF WATER? 2
WATER TEMPERATURE? 97
THE WATER TEMPERATURE IS HIGH ENOUGH TO DESTROY MOST LIFE.
TRY A NEW TEMPERATURE.
WATER TEMPERATURE? 22
YOUR BODY OF WATER IS A BLOCK OF ICE, AND CAN'T
ACCEPT ANY WASTE.  TRY A NEW TEMPERATURE.
WATER TEMPERATURE? 52
KIND OF WASTE? 3
KIND OF WASTE? 2
DUMPING RATE? 4
TYPE OF TREATMENT? 2

DO YOU WANT: A GRAPH(1), A TABLE(2), OR BOTH(3)? 3
```

Figure 3-12. Excerpt from POLUT program run.

```
TO SAVE TIME, I WILL SIMPLY PRINT ? AND YOU CAN TYPE
IN THE VALUE OF THE LIGHT INTENSITY YOU HAVE SELECTED.
AFTER YOU HAVE FINISHED TRYING DIFFERENT VALUES, TYPE
100 AFTER MY ? .

?67
USE ONLY VALUES BETWEEN 0 AND 60.
?OK.THANKS 54
P= 92
?0
P= 0
?-5
YOUR VALUE CAN NEVER BE LESS THAN 0
?OK, TRY 43
P= 122
?65
USE ONLY VALUES BETWEEN 0 AND 60.
```

Figure 3-13. Excerpt from PSYFC program run.

In the sample run of PSYFC,* Figure 3-13, look at the way the program handles inadmissable entries:

The program CRIMEX, from the Northwest Regional Educational Laboratory's URBAN CRIME UNIT,† is also designed to alert the student to input that is outside of the allowable range.

*Program used in REACT/Tecnica unit "Photosynthesis"; available from Tecnica Education Corporation.

†Urban Crime Unit, Computer Technology Program, Northwest Regional Educational Laboratory, Portland, Oregon, 1977.

```
DO YOU WANT TO CHANGE THE VALUE OF ANY OF THE CONDITIONS?YES

ENTER PRECINCT, ITEM NUMBER AND NEW VALUE.  FOR EXAMPLE IF YOU
WANT TO INCREASE THE NUMBER OF POLICE IN PRECINCT 3  TO A
TOTAL OF 240,  YOU WOULD ENTER  3,5,240  AFTER THE QUESTION
MARK IS PRINTED.  WHEN YOU HAVE NO MORE CHANGES TO MAKE,
ENTER  0,0,0
?1,1,2500
?2,2,9
 9     IS TOO DRASTIC!  MAKE CHANGE SMALLER.
?2,2,10.5
?3,9,6
THERE ARE ONLY EIGHT CONDITIONS THAT YOU MAY CHANGE.
?3,8,6
 6     IS TOO DRASTIC!  MAKE CHANGE SMALLER.
?3,8,6.7
?3,3,3
?6,1,3700
THERE ARE ONLY FIVE PRECINCTS IN THE CITY.
?5,1,3700
 3700     IS TOO DRASTIC!  MAKE CHANGE SMALLER.
?5,1,4250
?0,0,0
```

Figure 3-14. Excerpt from CRIMEX program run.

Some programs are designed to handle inappropriate or misspelled input by printing out responses such as:

WHAT?

or:

PLEASE ANSWER AGAIN, CORRECTLY

or:

EITHER YOU ARE GOOFING AROUND OR YOU NEED TO
PRACTICE YOUR TYPING. ANSWER THE QUESTION
AGAIN, TYPING CAREFULLY.

Many computers print out large numbers in "scientific notation." For example, the number 10,026,771,000,000 would be printed as:

10.026771E12

which is likely to confuse a student not familiar with the notation. There are, in fact, ways to write programs to avoid this kind of notation; if you can run the program yourself, be sure to check on the form used for large numbers. If they are printed in scientific notation in a social science unit, for example, the unit might very well be given a lower rating under student-proof.

Finally, a truly student-proof program will be written so that when students enter input in an incorrect form, try to divide by zero, and so forth, a message will be printed out explaining clearly what is wrong and how to correct the input error, so that students will not be confronted with confusing system messages like ?? or EXTRA INPUT–WARNING ONLY, which does not explain what mistake was made or how to correct it.

Now examine the sample units you are using in this course and make as many runs of the programs as you need to in order to determine how student-proof they are, rating each on the scale from 1 to 4:

1 = very poor
2 = poor
3 = good
4 = very good

Note your rating for each program on the decision table on page 105.

3. Extent to Which Unit is User-Oriented

The question of how user-oriented a unit is can best be answered by examining the input and output in a sample run. It should be made clear to the user what input is required and in what format. The output should be self-explanatory, brief, and succinct. A delicate balance is required to achieve clear, understandable output without burdening the user with endless time-consuming printout. Some examples of input and output follow, with brief discussions of their user-orientation features.

To start with, read through the complete sample run of a program called SPAWAR* (Figure 3–15). As you see, there is a great deal of information printed out before the program pauses to request the first input from the student. This introductory information takes four minutes to print out, with about half of that time being consumed in printing the diagram of the planet and the space ship. The length of this material suggests that the program would be more effective if it could be run on a display-type terminal, which outputs much faster than a teletypewriter. Even on a teletypewriter, however, time could be saved if the diagram of the planet were printed in the student's manual. In this case, the teacher could have the program revised to exclude the diagram.

You probably also noticed the lengthy repetition of the input questions throughout the run. Considerable printing time could be saved if the form of the question were shortened after the first iteration; the form of the computer's answers could also be shortened after the first response. Compare the original run with the shorter version shown in Figure 3–16, which runs in half the time. With SPAWAR, there will always be considerable time consumed (probably several minutes) each time the student must ponder and enter an angle and detonation distance. Compared to this total time, saving two or three minutes of printout may not make much difference.

One additional comment on SPAWAR's user-orientation. Notice that the explosion distances are printed out in the form:

```
88.916*10↑2
```

*From Space War unit, available from Project SOLO, University of Pittsburgh.

```
GET-SPAWAR
RUN
SPAWAR

SOMEWHERE ABOVE YOUR PLANET IS A ROMULAN SHIP.

THIS SHIP IS IN A CONSTANT POLAR ORBIT. IT'S
DISTANCE FROM THE CENTER OF YOUR PLANET IS FROM
10,000 TO 30,000 MILES AND AT IT'S PRESENT VELOCITY CAN
CIRCLE YOUR PLANET ONCE EVERY 12 TO 36 HOURS.

UNFORTUNATELY THEY ARE USING A CLOAKING DEVICE SO
YOU ARE UNABLE TO SEE THEM, BUT WITH A SPECIAL
INSTRUMENT YOU CAN TELL HOW NEAR THEIR SHIP YOUR
PHOTON BOMB EXPLODED. YOU HAVE SEVEN HOURS UNTIL THEY
HAVE BUILT UP SUFFICIENT POWER IN ORDER TO ESCAPE
YOUR PLANET'S GRAVITY.

YOUR PLANET HAS ENOUGH POWER TO FIRE ONE BOMB AN HOUR.

AT THE BEGINNING OF EACH HOUR YOU WILL BE ASKED TO GIVE AN
ANGLE (BETWEEN 0 AND 360) AND A DISTANCE IN UNITS OF
100 MILES (BETWEEN 100 AND 300), AFTERWHICH YOUR BOMB'S
DISTANCE FROM THE ENEMY SHIP WILL BE GIVEN.

AN EXPLOSION WITHIN 5,000 MILES OF THE ROMULAN SHIP
WILL DESTROY IT.

BELOW IS A DIAGRAM TO HELP YOU VISUALIZE YOUR PLIGHT.

                        90
                         !
                  0000000000000
               0000000000000000000
             000000           000000
           00000                 00000
          00000    XXXXXXXXXXX    00000
         00000   XXXXXXXXXXXXXXX   0000
        0000    XXXXXXXXXXXXXXXXX   0000
        0000   XXXXXXXXXXXXXXXXXXX   0000
       0000   XXXXXXXXXXXXXXXXXXXXX  0000
180<== 00000  XXXXXXXXXXXXXXXXXXXXX   00000 ==>0
       0000   XXXXXXXXXXXXXXXXXXXXX  0000
        0000   XXXXXXXXXXXXXXXXXXX   0000
        0000    XXXXXXXXXXXXXXXXX   0000
         00000   XXXXXXXXXXXXXXX   00000
          00000    XXXXXXXXXXXXX  00000
           00000                 00000
             000000           000000
               0000000000000000000
                  0000000000000000
                         !
                        270

X - YOUR PLANET
O - THE ORBIT OF THE ROMULAN SHIP

ON THE ABOVE DIAGRAM, THE ROMULAN SHIP IS CIRCLING
COUNTERCLOCKWISE AROUND YOUR PLANET. DON'T FORGET
WITHOUT SUFFICIENT POWER THE ROMULAN SHIP'S ALTITUDE
AND ORBITAL RATE WILL REMAIN CONSTANT.

GOOD LUCK. THE FEDERATION IS COUNTING ON YOU.

HOUR 1    , AT WHAT ANGLE DO YOU WISH TO SEND
YOUR PHOTON BOMB?
?0
HOW FAR OUT DO YOU WISH TO DETONATE IT?
?100

YOUR PHOTON BOMB EXPLODED 88.916     *10↑2 MILES FROM
THE ROMULAN SHIP

HOUR 2    , AT WHAT ANGLE DO YOU WISH TO SEND
YOUR PHOTON BOMB?
?1
HOW FAR OUT DO YOU WISH TO DETONATE IT?
?100

YOUR PHOTON BOMB EXPLODED 81.9322     *10↑2 MILES FROM
THE ROMULAN SHIP

HOUR 3    , AT WHAT ANGLE DO YOU WISH TO SEND
YOUR PHOTON BOMB?
?3
HOW FAR OUT DO YOU WISH TO DETONATE IT?
?100

YOUR PHOTON BOMB EXPLODED 76.2535     *10↑2 MILES FROM
THE ROMULAN SHIP

HOUR 4    , AT WHAT ANGLE DO YOU WISH TO SEND
YOUR PHOTON BOMB?
?5
HOW FAR OUT DO YOU WISH TO DETONATE IT?
?100

YOUR PHOTON BOMB EXPLODED 70.506     *10↑2 MILES FROM
THE ROMULAN SHIP

HOUR 5    , AT WHAT ANGLE DO YOU WISH TO SEND
YOUR PHOTON BOMB?
?9
HOW FAR OUT DO YOU WISH TO DETONATE IT?
?100

YOUR PHOTON BOMB EXPLODED 67.6104     *10↑2 MILES FROM
THE ROMULAN SHIP

HOUR 6    , AT WHAT ANGLE DO YOU WISH TO SEND
YOUR PHOTON BOMB?
?9
HOW FAR OUT DO YOU WISH TO DETONATE IT?
?105

YOUR PHOTON BOMB EXPLODED 61.7247     *10↑2 MILES FROM
THE ROMULAN SHIP

HOUR 7    , AT WHAT ANGLE DO YOU WISH TO SEND
YOUR PHOTON BOMB?
?9
HOW FAR OUT DO YOU WISH TO DETONATE IT?
?115

YOUR PHOTON BOMB EXPLODED 59.5767     *10↑2 MILES FROM
THE ROMULAN SHIP
YOU HAVE ALLOWED THE ROMULANS TO ESCAPE.
ANOTHER ROMULAN SHIP HAS GONE INTO ORBIT.
DO YOU WISH TO TRY TO DESTROY IT?
?NO
PLEASE LOGOUT

DONE
```

Figure 3–15. Sample run of SPAWAR.

```
HOUR     1, AT WHAT ANGLE DO YOU WISH TO SEND
YOUR PHOTON BOMB?
?0
HOW FAR OUT DO YOU WISH TO DETONATE IT?
?100

YOUR PHOTON BOMB EXPLODED 88.916    *10↑2 MILES FROM
THE ROMULAN SHIP

ANGLE FOR HOUR 2    ?1
DETONATE AT ?100

EXPLODED 81.9322    *10↑2 MILES AWAY

ANGLE FOR HOUR 3    ?3
DETONATE AT ?100

EXPLODED 81.9322    10↑2 MILES AWAY

ANGLE FOR HOUR 4    ?5
DETONATE AT ?100
```

Figure 3–16. Shortened segment for SPAWAR.

Some students may not know that this means:

8891.6,

Therefore, this numeric form should be explained in the student manual or the printout.

Now, read through the run of program WEATHR* shown in Figure 3–17. The printout looks relatively economical and clear. Notice, however, that there are no directions telling the user what form to use for input, such as "When entering the different statistics for the two days, enter one number, use a comma, then enter the second number." In the sample run, the user seemed to use the correct form automatically when entering temperature and barometric pressure, but look at lines 9–11:

```
        BAROMETER TENDENCY (1=RISING, 2=FALLING, 3=STEADY);
?1
??1
```

The user entered the barometric tendency for only day, 1. The computer, needing a second entry, printed out two question marks. The user responded to the computer's ?? by repeating the first entry, 1. At this point, the computer has two numbers which it interprets as the tendencies for two days, but the student assumes he has entered just one number.

When evaluating user-orientation features, check to see that all needed directions are given for inputting data.

Looking back at the WEATHR printout, what do you think of the format for the forecast? Is it easy to read? Is it too long? Does it provide just the forecast information that might be necessary, or does it show too little or too much?

When a program generates a great deal of information, particularly as extensive tables, it can often be made more usable by providing options as to how much information is provided. In the

*The unit materials for this program are available from Minnesota Educational Computing Consortium.

```
GET-WEATHR
RUN
WEATHR

THIS PROGRAM WILL ATTEMPT TO PREDICT TOMORROWS WEATHER IF
GIVEN THE WEATHER STATISTICS FROM THE PAST TWO DAYS.
     SEASON
?SUMMER
     TEMPERATURE
?76,78
     BAROMETER
?29,29.5
     BAROMETER TENDENCY(1=RISING,2=FALLING,3=STEADY):
?1
??1
     RELATIVE HUMIDITY
?25,30
     CLOUDS(1=STRATUS,2=CUMULUS,3=CIRRUS)
?1
??1
     CLOUD COVER(PERCENTAGE)
?35,30
     WIND DIRECTION (1=NORTH, 2=SOUTH, 3=EAST, 4=WEST)
?2,2
     WIND SPEED
?5,10


-----------------------------------------------------
PRESENT SEASON IS SUMMER

FORECAST FOR TOMORROW:


TEMPERATURES:
LOWS TONIGHT BETWEEN 47   AND 57   DEGREES
HIGHS TOMORROW NIGHT BETWEEN 77   AND 82   DEGREES
LOWS TOMORROW NIGHT BETWEEN  42   AND 47   DEGREES
BAROMETER 29    AND RISING.
HUMIDITY BETWEEN  27.5        AND 32.5       PERCENT
CLOUD COVER BETWEEN 32.5         AND 37.5       PERCENT
CLOUD HEIGHT BETWEEN 500 TO 580 FEET.
MAJOR CLOUD TYPE WILL BE STRATUS.
WIND FROM THE SOUTH FROM 7.5          TO 12.5       MPH
CHANCE OF PRECIPITATION:
TONIGHT 91   %
TOMORROW 48   %
TOMORROW NIGHT 100   %

FORECAST FOR TOMORROWS WEATHER:

IT SHOULD BE FAIR TOMORROW.
IT SHOULD BE WARMER TOMORROW WITH NO PRECIPITATION LIKELY.

DONE
```

Figure 3–17. Run of WEATHR.

sample run of BANK* in Figure 3–18, you can see the effective use of options, first (A) to control the particular kind of calculations the program will perform, then (B) to control the amount of data printed out.

Another convenient user-oriented device is illustrated by the CIVIL† program in Figure 3–19, where the user has the option of seeing the rather long description of the program or of skipping it. Another time-saving device in CIVIL is that the listing of the 14 battles has not been made a part of the program run, but has been presented in the student materials accompanying the program.

*The unit materials are available in "Huntington I Application Programs–Mathematics," from Digital Equipment Corporation.

†Unit materials for CIVIL are available from Minnesota Educational Computing Consortium.

```
RUN
BANK

                        FINANCIAL PROBLEMS

THIS PROGRAM SOLVES THREE TYPES OF PROBLEMS:

A→      (1) INTEREST ON INSTALLMENT BUYING
        (2) PAYMENTS ON LONG TERM LOAN
        (3) BALANCE OF A SAVINGS ACCOUNT

WHICH PROBLEM WOULD YOU LIKE TO WORK WITH (TYPE 1, 2 OR 3)?1

*****

THIS SECTION WILL DETERMINE THE ACTUAL INTEREST YOU PAY
WHEN YOU PURCHASE SOMETHING ON CREDIT.

WHAT IS THE CASH PRICE OF THE ARTICLE ($)?2809.95
            DOWN PAYMENT ($)? 1500
            NUMBER OF PAYMENTS EXCLUDING THE DOWN PAYMENT?25
            NUMBER OF PAYMENTS PER MONTH? 1
            AMOUNT PER PAYMENT ($)?36.35


THE RATE OF INTEREST CHARGED WAS-47.35        PERCENT.


*****

WOULD YOU LIKE TO RUN THE PROGRAM AGAIN (1-YES, 0-NO)?1
WHICH PROBLEM WOULD YOU LIKE TO WORK WITH (TYPE 1, 2 OR 3)?2

*****

THIS SECTION WILL DETERMINE PAYMENTS FOR A LONG TERM LOAN.

WHAT IS THE AMOUNT BORROWED ($)?5500
            INTEREST CHARGED (%)?8
            INTERVAL BETWEEN PAYMENTS (MONTHS)?1
            TERM OF THE LOAN (YEARS)?2

B→DO YOU WISH TO SEE THE TOTALS ONLY - INSTEAD OF THE ENTIRE
TABLE - (1-YES, 0-NO)?0
```

PERIOD	OUTSTANDING PRINCIPAL AT BEGINNING OF PERIOD	INTEREST DUE AT END OF PERIOD	PRINCIPAL REPAID AT END OF PERIOD
1	5500	36.67	212.08
2	5287.92	35.25	213.5
3	5074.42	33.83	214.92
4	4859.5	32.4	216.35
5	4643.15	30.95	217.8
6	4425.35	29.5	219.25
7	4206.1	28.04	220.71
8	3985.39	26.57	222.19
9	3763.21	25.09	223.66
10	3539.55	23.6	225.15
11	3314.4	22.1	226.65
12	3087.75	20.59	228.17
13	2859.58	19.06	229.69
14	2629.89	17.53	231.22
15	2398.67	15.99	232.76
16	2165.91	14.44	234.31
17	1931.6	12.88	235.87
18	1695.73	11.3	237.45
19	1458.28	9.72	239.03
20	1219.25	8.13	240.62
21	978.63	6.52	242.23
22	736.4	4.91	243.84
23	492.56	3.28	245.47
24	247.09	1.65	247.1
TOTALS		470.	5500

```
YOUR MONTHLY PAYMENT IS $ 248.75       AND TOTALS $ 5970
```

Figure 3-18. Sample run of BANK.

```
RUN
CIVIL

DO YOU WANT DESCRIPTIONS?YES


THIS IS A CIVIL WAR SIMULATION.
TO PLAY, TYPE A RESPONSE WHEN THE COMPUTER ASKS.
REMEMBER THAT ALL FACTORS ARE INTERRELATED AND THAT YOUR
RESPONSES COULD CHANGE HISTORY. FACTS AND FIGURES USED ARE
BASED ON THE ACTUAL OCCURRENCE. MOST BATTLES TEND TO RESULT
AS THEY DID IN THE CIVIL WAR, BUT IT ALL DEPENDS ON YOU

THE OBJECT OF THE GAME IS TO WIN AS MANY BATTLES AS POSSIBLE

YOUR CHOICES FOR DEFENSIVE STRATEGY ARE:
        (1) ARTILLERY ATTACK
        (2) FORTIFICATION AGAINST FRONTAL ATTACK
        (3) FORTIFICATION AGAINST FLANKING MANEUVERS
        (4) FALLING BACK
 YOUR CHOICES FOR OFFENSIVE STRATEGY ARE:
        (1) ARTILLERY ATTACK
        (2) FRONTAL ATTACK
        (3) FLANKING MANEUVERS
        (4) ENCIRCLEMENT
YOU MAY SURRENDER BY TYPING A5 FOR YOUR STRATEGY.

YOU ARE THE CONFEDERACY. GOOD LUCK

SELECT A BATTLE BY TYPING A NUMBER FROM 1 TO 14 ON
REQUEST. TYPE ANY OTHER NUMBER TO END THE SIMULATION.

WHICH BATTLE DO YOU WISH TO SIMULATE? 1


THIS IS THE BATTLE OF BULL RUN
JULY21, 1861 GEN. BEAUREGARD COMMANDING THE SOUTH MET THE
UNION FORCES WITH GEN. MCDOWELL IN A PREMATURE BATTLE AT BULL
RUN. GEN. JACKSON HELPED PUSH BACK THE UNION ATTACK.
```

Figure 3–19. Excerpt from a sample run of CIVIL.

As a final example, read through the run of program SSINS* in Figure 3–20.

First, the directions for running this program and inputting data are clear and thorough. The device of giving examples to illustrate directions is effective for precluding errors. The incomprehensible codes used in the program are acceptable in this instance because the program merely does the calculations for the Inter-Nation Simulation. The materials for the Simulation provide the necessary background information and all the abbreviations will be familiar to the students working with it.

Now, considering the style of the program, what did you think of the way the computer responded to the user? Did it seem somewhat coy or condescending? In consideration of the user, and of time, programs usually do well to omit cute or clever remarks which, though fine for some users, may offend others. In addition, it is probably wise not to lead students to believe computers have personality, emotions, or a sense of humor.

As an exercise, rewrite the first section of the SSINS output down to the first data entry (line 16).

Examine the sample units you are using and rate each program on its user-orientation. Record your rating, from 1 to 4, on the decision table on page 105.

*The SSINS program was developed by Southern Minnesota Educational Computing Consortium.

```
RUN
SSINS

WHAT IS YOUR NAME ?MICHELLE
DO YOU WANT INSTRUCTIONS MICHELLE?YES
I, UNIVAC 1100, WILL HELP YOU MAKE THE CALCULATIONS THAT
YOU WILL NEED TO PLAY INS NEXT PERIOD.  YOU MUST HAVE YOUR
MAIN DECISCION FORM (MDF) COMPLETED AND WITH YOU NOW.
HAVE IT HANDY BECAUSE I DONT LIKE TO WASTE TIME. BRING
THE PRINT OUT TO CLASS AND HAVE YOUR MOST EXCELLENT AND
OUTSTANDING TEACHER O.K. IT BEFORE YOU BEGIN THE NEXT
SESSION.  DO NOT USE COMMAS WHEN INPUTTING NUMBERS IN
THOUSANDS.  EXAMPLE-- 20000
ENTER ALLL PERCENTAGES AS DECIMALS.  EXAMPLE...ENTER 1.7%
AS THE DECIMAL .017.
ARE YOU READY ?YES
GOOD!  WE SOCIAL SCIENTISTS ALWAYS TRY TO BE READY!
ENTER DL, FCC, FCN, BC-DF-BC, BC-DF-N, AND BR FROM MDF
?4,5000,410,50,300,.95
NOW ENTER POP, %POP, UC, DR, BC, PERIOD, AND NATION
FROM MDF
?183.7,.017,297.2,.10,51870,1,BINGO
NOW ENTER BUDGET ALLOCATIONS FROM THE MDF, CI%, CS%
FC%, RD%, AND DL%
?.14,.73,.1,.03,0
NEXT ENTER BR-IMP,BR-EXP,UC-IMP,UC-EXP,CS-IMP,CS-EXP,
FROM THE MDF
?.01,0,0,3,0,0
NEXT ENTER FCC-IMP,FCC-EXP,FCN-IMP,FCN-EXP
?0,0,0,0
BINGO HAS  5187      BC S TO ALLOCATE FOR FC
ENTER AMOUNTS IN BC FOR FCC, FCN,BC-DEF-BC,BC-DEF-N
?2500,2500,187,0
ENTER NUMBER OF FCN PRODUCED WITH  2500      BC S
?625

THESE ARE THE CHARECTERISTICS OF BINGO FOR PERIOD  2


DS=                   20.6125
DL=                   4
FCC=                  7000
FCN=                  625
BC DEF BC=            234.5
BC DEF N=             285
BR=                   .96
POPULATION=           186.823
POP. GROWTH RATE=     .017
UC=                   305.968
DR=                   .1
DOES BINGO HAVE A LOAN FROM THE W.B. ...1=YES,0=NO?0
BC=                  54575.4
DO YOU HAVE ANYMORE MDF S TO FIGURE...1=YES,2=NO? 2
IT HAS BEEN FUN WORKING WITH YOU MICHELLE.  GOODLUCK
IN THE NEXT INS SESSION!

DONE
```

Figure 3-20. Sample run of SSINS.

4. Variables Controllable by User

It is important, particularly in learning experiences designed to promote "discovery," that the user be able to manipulate the critical variables in the program. Especially with simulations and problem-solving units, the student should be able to ask freely the question "what if . . ." and change the values of variables to explore that question. In addition, the teacher ought to be able to change the value of the parameters easily, or set limits of the program, to accommodate the varying needs and abilities of students.

If a program is a rigid, inflexible package, which the user can interact with only in a predetermined way, its usefulness as a teaching tool is severely limited. An example of the opposite is BALPAY,* a sample run of which is shown in Figure 3–21. The student is asked to control the values of four variables:

1. Taxes on investments abroad;
2. Tariff rate;
3. Government spending abroad;
4. Prime interest rate.

By varying the values, the student is to discover the relationships between the variables and the International Balance of Payments. (Note: Wisdom and efficiency would suggest that only one variable at a time be manipulated, with all others remaining constant, in order to get a clear picture of the effect of that variable in the model.) The teacher may change some limits of the program to make the game easier or more difficult, depending on the sophistication of the students using it. The teacher may change:

1. The number of years in the time limit;
2. The value of the ratio used to determine win/lose (set at +.03 in the example).

Rate the unit BALPAY according to the number and the utility of variables that are directly controllable by the user.

For further practice in evaluating the use of variables in typical units, obtain sample program runs for the units you are using with this course. Rate each one's use of variables on the 1 to 4 scale and record your rating on a decision table like the one on page 105.

5. Options for Use

Often, a single unit is more useful to a teacher if it can be used to achieve several different (but related) sets of objectives, or if it can be used in more than one setting. Examine the materials for the URBAN CRIME UNIT (or some other unit) to determine what options are offered, then compare the available options with those of the other units you are using in this course. Rate each on its use of options and record the ratings on your decision table.

6. Time and Cost of Using

Time and cost are multifaceted variables. First, there is the straightforward central processor time used in running the program. This figure can usually be obtained by running the program and then signing off the terminal. The computer should automatically print the CPU time (in seconds) used before signing off. From this figure it is usually fairly easy to compute the cost of

*Available in "REACT Social Science Application Units," from Tecnica Education Corporation.

```
RUN
BALPAY

DO YOU WANT A DESCRIPTION OF THIS GAME?YES

THIS COMPUTER PROGRAM IS A DECISION MAKING GAME.
YOUR PART IN THE GAME IS THAT OF DECISION MAKER
FOR THE COUNTRY.  ASSUME THAT THE COUNTRY IS CURRENTLY
IN A VERY POOR BALANCE-OF-PAYMENTS-POSITION. YOUR
OBJECTIVE IS TO MAKE DECISIONS THAT WILL GIVE THE
COUNTRY A HEALTHY BALANCE OF PAYMENTS POSITION WITHIN  4 YEARS.

BALANCE STATEMENT IN MILLIONS OF DOLLARS - YEAR 0

INFLOWS

EXPORTS      GOODS & SERVICES      $ 19 073
EXPENDITURES-FOREIGN TOURIST       $  2 153
INCOME       INVESTMENT            $    902

NET INFLOW   $ 22128

OUTFLOWS

IMPORTS      GOODS & SERVICES      $ 18 271
EXPENDITURES TOURISTS ABROAD       $  1 610
FOREIGN AID & MILITARY AID         $  4 745
INVESTMENTS FOREIGN                $  2 375

NET OUTFLOW   $ 27001

NET SURPLUS(+) OR DEFICIT(-)   -4873

NET SURPLUS OR DEFICIT/NET INFLOW   -.23

AS YOU CAN SEE FOR THE STATEMENT ABOVE, OUTFLOWS
EXCEED INFLOWS BY A SUBSTANTIAL AMOUNT. YOUR COUNTRY
IS IN A SERIOUS DEFICIT POSITION.  SHOULD THE DEFICIT
CONTINUE FOR LONG, YOUR COUNTRY'S GOLD RESERVES WOULD
BE DEPLETED. ALSO TOO MUCH OF YOUR COUNTRY'S RESOURCES
AND PRODUCTION ARE PROBABLY BEING SENT OUT OF THE COUNTRY

FOR THE PURPOSES OF THIS GAME, A 'HEALTHY' BALANCE OF
PAYMENTS POSITION IS DEFINED AS ONE IN WHICH THE VALUE
OF THE RATIO, 'NET SURPLUS OF DEFICIT/NET INFLOW' IS
BETWEEN THE VALUES OF -.03        AND  .03       . NOTICE
THAT THE CURRENT VALUE OF THIS RATIO IS -.23

ENGLAND HAS REVALUED ITS CURRENCY.  THE ENGLISH GOVERNMENT
WILL NOW PURCHASE AND SELL GOLD FOR 5 SHILLINGS MORE AN
OUNCE. THIS MOVE IN EFFECT HAS INCREASED THE VALUE OF YOUR
CURRENCY RELATIVE TO ENGLAND'S. THEREFORE GOODS MANUFACTURED
IN YOUR COUNTRY ARE NOW MORE EXPENSIVE TO THE ENGLISH
PEOPLE AND ENGLISH GOODS ARE LESS EXPENSIVE TO YOU.
```

```
ENTER YOUR POLICY DECISIONS FOR YEAR  1

WHAT PERCENT CHANGE DO YOU WISH
IN TAXES ON INVESTMENTS ABROAD?3
IN THE TARIFF RATE?4.5
IN GOVERNMENT SPENDING ABROAD?-5
IN THE PRIME INTEREST RATE?.5

BALANCE STATEMENT - YEAR 1

INFLOWS

EXPORTS       $ 18 099
EXPENDITURES  $  2 112
INCOME        $    870

NET INFLOW   $ 21081

OUTFLOWS

IMPORTS       $ 15 746
EXPENDITURES  $  1 598
FOREIGN AID   $  4 507
INVESTMENTS   $  1 899

NET OUTFLOW   $ 23750

$ DIFFERENCE  -2669

% DIFFERENCE  -.13

THERE HAS BEEN A REVOLUTION IN PERU.  THE NEW
GOVERNMENT HAS TAKEN OVER ALL OF THE INDUSTRY CONTROLLED
BY FOREIGN INVESTORS.  ALSO HARSH RESTRICTIONS HAVE BEEN
IMPOSED ON IMPORTS FROM YOUR COUNTRY.

ENTER YOUR POLICY DECISIONS FOR YEAR  2

WHAT PERCENT CHANGE DO YOU WISH
IN TAXES ON INVESTMENTS ABROAD?-5
IN THE TARIFF RATE?0
IN GOVERNMENT SPENDING ABROAD?5
IN THE PRIME INTEREST RATE?-.4

BALANCE STATEMENT - YEAR 2

INFLOWS

EXPORTS       $ 18 165
EXPENDITURES  $  2 120
INCOME        $    901

NET INFLOW   $ 21186
```

Figure 3-21. Sample run of BALPAY (continued on next page).

```
OUTFLOWS

IMPORTS        $ 15 958
EXPENDITURES   $  1 629
FOREIGN AID    $  4 826
INVESTMENTS    $  2 429

NET OUTFLOW    $ 24842

$ DIFFERENCE   -3656

% DIFFERENCE   -.18

BECAUSE OF THE RELAXING OF TENSIONS IN THE MIDDLE EAST
YOUR COUNTRY HAS DECIDED TO BRING HOME 75,000 TROOPS
NOW STATIONED OUTSIDE OF THE COUNTRY.


ENTER YOUR POLICY DECISIONS FOR YEAR  3

WHAT PERCENT CHANGE DO YOU WISH
IN TAXES ON INVESTMENTS ABROAD?21

YOUR POLICY DECISION IS TOO DRASTIC AND WOULD
CAUSE SERIOUS ECONOMIC PROBLEMS IN YOUR COUNTRY

ENTER ANOTHER DECISION.

IN TAXES ON INVESTMENTS ABROAD?6
IN THE TARIFF RATE?-5
IN GOVERNMENT SPENDING ABROAD?-7
IN THE PRIME INTEREST RATE?0

BALANCE STATEMENT - YEAR 3

INFLOWS

EXPORTS        $ 18 471
EXPENDITURES   $  2 125
INCOME         $    851

NET INFLOW    $ 21447

OUTFLOWS

IMPORTS        $ 18 335
EXPENDITURES   $  1 564
FOREIGN AID    $  4 353
INVESTMENTS    $  1 735

NET OUTFLOW    $ 25987

$ DIFFERENCE   -4540

% DIFFERENCE   -.22
```

```
ENTER YOUR POLICY DECISIONS FOR YEAR  4

WHAT PERCENT CHANGE DO YOU WISH
IN TAXES ON INVESTMENTS ABROAD?6
IN THE TARIFF RATE?2
IN GOVERNMENT SPENDING ABROAD?0
IN THE PRIME INTEREST RATE?-3

BALANCE STATEMENT - YEAR 4

INFLOWS

EXPORTS        $ 18 123
EXPENDITURES   $  2 108
INCOME         $    809

NET INFLOW    $ 21040

OUTFLOWS

IMPORTS        $ 17 074
EXPENDITURES   $  1 537
FOREIGN AID    $  4 353
INVESTMENTS    $  1 192

NET OUTFLOW    $ 24156

$ DIFFERENCE   -3116

% DIFFERENCE   -.15

UNFORTUNATELY YOU HAVE NOT RESTORED THE COUNTRY TO
A HEALTHY BALANCE OF PAYMENTS POSITION IN THE TIME
ALLOWED.  YOU SHOULD RETHINK YOUR STRATEGY BASED ON
THIS EXPERIENCE AND TRY THE GAME AGAIN.

DO YOU WANT TO TRY THE GAME AGAIN NOW?NO

DONE
```

Figure 3-21. Continued.

computer time involved in running the program. The cost per second, which varies from computer to computer, can be furnished by your local computer expert. A typical CPU time is illustrated below:

```
RUN
IPS2.2

STUDENT          SAND          AIR          RATIO
1                18.8          15.2         .45
2                20            15           .43
3                18            13           .42
4                18.6          15.4         .45
30               18.5          13.5         .42
31               19            17           .47
32               21            15           .42
33               23            13           .36

AVERAGE RATIO    .44
MEAN DEVIATION   .025

TIME: 0.439 SEC.
READY
```

Figure 3–22. Printout showing CPU time used.

In addition to CPU time, you will be concerned with terminal time—that is, the amount of time students must spend at the terminal to make the required program runs. Some unit materials specify this; if not, you can find out by running the program yourself. Besides these simple figures, you will want to estimate the teacher and class time a unit will involve, including your preparatory time; the class time involved in preparatory activities; actual use of the unit; and follow-up activities. You may be able to determine the dollar cost of this time.

To evaluate the time and cost factors, you will probably find it sufficient to add together the computer, terminal, and teacher-class time (and cost). You can then reassess whether these are reasonable and whether you have enough time and money to use the unit. If the totals are inordinately high compared to other methods of achieving your instructional objectives, you may very well rank the unit quite low.*

Making the Final Decision

Once you have rated a unit on each of these six criteria plus those you have added, you are ready to make your final decision. In most cases, even with all the quantitative data you have collected, the ultimate decision still will be largely subjective.

For example, you may have the rating record shown in Figure 3–23 for units that teach one particular skill. Although Unit 6 has the highest total points, you might decide to use Unit 2 instead because of its superior enhancement qualities and its better time/cost rating. Or, you might

*Until you can check with someone familiar with your school's computer system, you will probably not be able to estimate time and cost for the sample units you are using with this course; therefore, omit trying to rate units on this final variable for now.

INITIAL DECISION VARIABLES	POTENTIALLY USABLE UNITS						
	#1	2	3	4	5	6	7
1. Will program run on hardware?	NO	YES	YES	YES	YES	MAYBE	MAYBE
2. Is language available?		YES	MAYBE	YES	NO	YES	MAYBE
3. Are objectives consistent with mine?		MAYBE	MAYBE	MAYBE		YES	MAYBE
4. Are materials complete enough?		YES	YES	NO		MAYBE	MAYBE
5. Is model appropriate?		YES	NO			MAYBE	MAYBE
PRELIMINARY DECISION	REJECT	EVALUATE FURTHER	REJECT	REJECT	REJECT	EVALUATE FURTHER	EVALUATE FURTHER
FINAL DECISION VARIABLES							
Enhancement		4				3	1
Student-proof		1				4	1
User-oriented		2				3	1
Variables		3				2	2
Options		2				3	1
Time/Cost		3				2	1
TOTALS		15				17	7
FINAL DECISION		?				?	REJECT

Figure 3-23. Sample preliminary and final decision tables for seven units.

feel that the student-proof and user-oriented qualities of the Unit 2 program are just as important as enhancement, time, and money.

To help you make close decisions in a more objective way, you may go one step further in your selection procedure by assigning weights to the evaluation criteria, according to your own preferences and priorities. Decide the relative importance to you of each variable and then assign to each a value that reflects the importance of the criterion. A simple way to do this would be to assign weights as percentages, totaling 100%. The weighted score for each unit will not only reflect the unit's strengths and weaknesses but also reflect the unit's relative worth to you according to your own priorities.

Figure 3-24 is a sample decision table for selecting your own computer-based instructional units.

INITIAL DECISION VARIABLES	UNIT NAME	UNIT NAME	UNIT NAME	UNIT NAME	UNIT NAME
1. Will program run on hardware?					
2. Is language available?					
3. Are objectives consistent with mine?					
4. Are materials complete enough?					
5. Is model appropriate?					
PRELIMINARY DECISION					
FINAL DECISION VARIABLES					
Enhancement					
Student-proof					
User-oriented					
Variables					
Options					
Time/Cost					
TOTALS					
FINAL DECISION					

Figure 3-24. Sample decision table.

IV

READINGS IN COMPUTER IN THE CURRICULUM

PREFACE

The articles comprising Chapter 4 were written especially for this Computer Technology Program course by authors experienced in using and/or developing computer-based instructional applications. Each article discusses the current instructional applications of the computer in the author's subject area, with the primary focus in most cases on the first five instructional modes described in Part I: drill, tutorial, problem-solving, data analysis, and simulation. Each article concludes with a complete bibliography.

The articles reflect computer usage in many subject areas, but they by no means exhaust the range of use in the modern curriculum. In all, they provide a synoptic view of computers in classrooms today which suggests the exciting avenues this technology is opening to students and teachers alike.

INTRODUCTION TO COMPUTERS IN THE CURRICULUM
by Daniel L. Klassen

BACKGROUND

In 1969 the American Institute for Research (AIR) conducted a survey of secondary schools in the United States to determine the current and anticipated future use of computers as administrative and instructional tools.* The study found that approximately 15 percent of the schools surveyed used computers in the instructional process, with the majority concentrated in the Western, North Central, and Northeastern regions and Oklahoma and Texas in the South. A follow-up study by AIR in 1975† found the concentration to be essentially the same, with the largest numbers clustered in the Western, North Central, and Northeastern sections and in Texas, as shown on the map in Figure 4-1.

The results of both surveys were used to project future use. The 1975 projection, shown in Figure 4-2, estimates that by 1984 half of the secondary schools in the United States will be using computers for instructional purposes.

There is considerable evidence to suggest that this projection is accurate. One measure is the increased sale of computers to schools. All of the computer manufacturers that attempt to sell to the educational market report expanding sales in the secondary school market. Much of this increase is due to the reduction in cost of the minicomputer, and more recently the microcomputer, making it possible for a single school or school district to lease or purchase a computer.

Another measure is the increase in the number

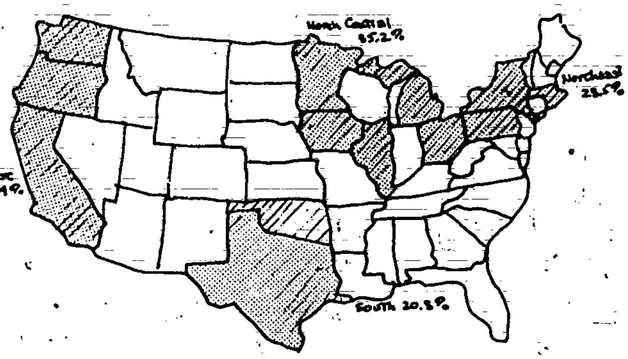


Figure 4-1. Geographic distribution of instructional users in 1970 (area shaded by lines) and in 1975 (area shaded by dots).

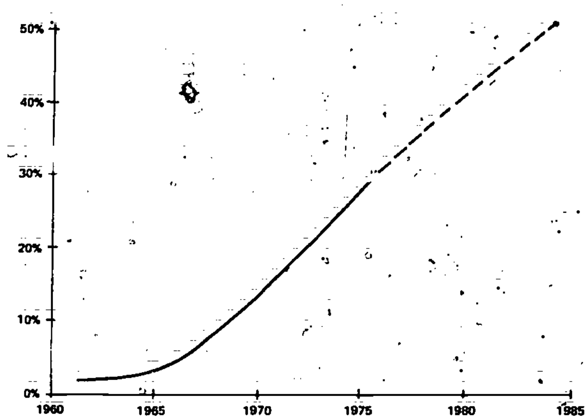


Figure 4-2. Growth of the use of instructional computers in secondary education.

*Charles A. Darby, Jr., Arthur L. Korotkin, and Tania Romashko, *The Computer in Secondary Schools: A Survey of its Instructional and Administrative Usage.* New York: Published in cooperation with the American Institutes for Research, Washington Office, by Praeger Publishers, 1972.

†William J. Bukoski and Arthur L. Koratkin, *Computer Activities in Secondary Education: Final Report,* American Institute for Research, September 1975.

	No. of Responses	Math.	Comp. Sci. Courses	Science	Voc/Ed Guid.	Social Science	Bus. Ed.	Lang.	Other	Indep Work	Total
CAI	222	59.0%	3.6%	17.6%	---	3.6%	3.2%	8.1%	4.5%	.4%	100.0%
Problem Solving	445	62.3	2.7	27.3	---	1.3	2.9	.4	2.0	1.1	100.0
Computer Science	432	29.6	61.6	1.6	.7%	.2	4.4	---	.7	1.2	100.0
Gaming Simulation	235	30.6	11.9	26.4	---	17.4	1.7	1.3	4.3	6.4	100.0
CMI	43	48.8	---	16.3	---	16.3	2.3	14.0	2.3	---	100.0
Guidance Counseling	79	---	---	---	96.2	---	---	---	2.5	1.3	100.0

Figure 4-3. Types of computer application by subject matter, 1975*.

of time-sharing cooperatives and networks. A time-sharing network consists of a number of schools sharing the same computer via telephone lines. Time-sharing networks have been organized by computer manufacturers, by independent commercial organizations, and by the school districts themselves.

Probably the most obvious measure is the growth in the development and availability of computer-based instructional applications. A large volume of computer-related curriculum materials has been produced by federally funded projects, by the computer manufacturers, by private publishers, by college and university staff, and by individual teachers.

Along with the increased availability of computers in the schools has gone an expansion of the subject areas using the computer to support and enhance instruction. Figure 4-3 shows the distribution of curriculum applications across subject areas. These 1975 AIR survey data indicate that while the most extensive use of the computer continues to occur in mathematics and science, the computer appears in restriction in a wide variety of other subject areas.

The Future

It is probably still true that most educators are unprepared to deal with the present realities of computer technology in education, much less to evaluate the probable future realities or to participate in designing alternatives or defining what the computer should become for education. Anthony Oettinger, in his book, *Run, Computer, Run*, writes that "massive education and re-education will become necessary to produce a breed of teachers who can feel at home with technical devices."† The need for education of educators with respect to computer technology was also pointed up by a George Washington University study of problems associated with the effects of computer technology on education, which recommended that "schools of education and other teacher training institutions and agencies be assisted to improve their familiarity with and investment in the use of the computer in education."‡ Dr. Sylvia Charp made a similar point before the IFIP World Conference on Computer Education when she said:

> One problem facing designers of educational information systems is that educators need to become better informed to understand and effectively apply this emerging technology . . . . In the hands of *knowledgeable* people, today's computer becomes a powerful tool that can be used to handle problems of great volume and complexity in attacking educational problems. Educators *must begin* to understand the capabilities of the computer, and plan ahead for its use.**

*Bukoski and Korotkin, p. 25.

†Anthony Oettinger, with Sema Marks, *Run, Computer, Run: The Mythology of Educational Innovation.* New York: Collier Books, 1969.

‡*Education in the 70's: A Study of Problems and Issues Associated with the Effects of Computer Technology on Education.* Washington, D.C.: George Washington University, 1967.

**Sylvia Charp, "Computer Technology in Education—How to Make it Viable." World Conference on Computer Education. IFIP 1970.

In the case of educators, ignorance of computer technology breeds vulnerability and places the destiny of education at the mercy of the technologists. There then enters the risk that the technology will shape education, whereas in fact education should shape the technology to suit its needs.

It should be emphasized that the need is not simply one for training educational personnel to use whatever technology comes down the educational road; the inherent dangers in that approach could be fatal for education. Instead, educators must learn to be active participants in the decisions about both the "becoming" of technology and its implementation. As Emmanuel Mesthene observed, "What is good for the educational technologists is not necessarily good for education." It is the job of the educators to determine what *is* good for education and then to bring technology to bear on that determination, if and where it is appropriate to do so. But the technologists are already knocking on the schoolhouse door, and unless the educators become educated, they will be unable either to evaluate the uses competently or to use them appropriately.

BIBLIOGRAPHY

1. Braun, Ludwig, "Digital Simulation in Education." *Journal of Educational Technology Systems, 1,* (1972), 5-27.
2. Charp, Sylvia, "Computer Technology in Education—How to Make it Viable." *World Conference on Computer Education.* 1970.
3. Darby, Charles A., Jr., Korotkin, Arthur L., and Romashko, Tania, *The Computer in Secondary Schools: A Survey of its Instructional and Administrative Usage.* New York: Published in cooperation with the American Institutes for Research, Washington Office, by Praeger Publishers, 1972.
4. *Education in the 70's: A Study of Problems and Issues Associated with the Effects of Computer Technology on Education.* Washington, D.C.: George Washington University, 1967.
5. Koch, Warren J., *The Use of Computers in Instruction in Secondary Schools.* Washington, D.C.: National Association of Secondary School Principals, 1972.
6. Oettinger, Anthony, with Marks, Sema, *Run, Computer, Run: The Mythology of Educational Innovation.* New York: Collier Books, 1969.

COMPUTERS IN ART EDUCATION by Peter Lynn Sessions

INTRODUCTION

Instruction in art is designed to foster the development of specific artistic skills and techniques; impart knowledge of important terms, concepts, and historical developments in art; and enhance the appreciation of works of art.

The development of such skills as pencil drawing or welding requires knowledge (how to select materials, for example), frequent chances to observe the work of others, timely guidance, and much practice. The student's goal is to master or explore the effects available with a given medium, learning at the same time to express a personal style. The goals of the teacher in this area might be to encourage students to experiment, and to transmit the techniques and standards of excellence of previous master artists and teachers.

Knowledge of terms and concepts in art (perspective, still life, geodesic) and of historical developments (chiaroscuro, impressionism, Brancusi) is acquired through direct visual experience and discussion relating the work of art to concepts and historical patterns. The student may wish to master or encounter the language of art, art history, and/or art criticism; it includes many terms, names, conceptual structures. In the realm of art instruction, the teacher's goal might be to provide as wide a selection of important areas of art knowledge as possible and present them in ways that facilitate exploration and mastery.

Art appreciation includes instruction in the areas of skills and knowledge discussed above; an awareness of how a work of art was produced (technique), together with a knowledge of the artist and period (terms, concepts, history), are part of what is commonly meant by appreciation. Also included might be studies of the physiological and psychological bases of visual experience: figure/ground relationships, closure, contrast, apparent size, and so on. The student's goal might include the development or justification of a preference for certain styles or artists, through an understanding of the processes of art. The teacher's goal might be to provide the student with a wide and comprehensive basis for judging works of art, including the student's own productions.

In summary, a practical art curriculum develops skills through personal guidance and opportunity for practice. Art instruction also seeks to place before the student a wealth of direct visual images and a number of symbols (terms, concepts, and so on) with which to interpret the images.

Let us now consider some of the features of the computer that might suit it for the art curriculum. The computer is a device for manipulating symbols—numbers, letters, words. The user must learn to encode a symbolic representation of his problem or to use representation encoded by someone else. In addition, the user needs to learn how to decode the results.

The symbolically encoded form of a problem or process is a program. Computers in common use in education today contain a special memory to store programs. Several types of educational programs have been described in Chapter 2.

Drill and practice programs
Tutorial programs
Problem-solving programs
Data-processing programs
Simulations or games

We shall apply each of these program roles to art.

The input/output devices (or peripherals) are of especial interest in the art curriculum, since they provide the means for translating the images which are a basis of art instruction into the symbols to be used by the computer. The output device determines the quality of the image which the computer process may directly produce. The production of direct images (computer art) is an important area for development in art instruction.

The Teletype terminal may make use of its characters (capital letters, numbers, many special characters) to form images. (The standard Teletype output is like that of a pica typewriter—10 characters to the inch horizontally, 6 lines to the inch vertically.)

The cathode-ray tube (CRT), essentially like a TV screen, displays user input and computer output. The format of the display is just like the teletype or pica typewriter and the allowable characters are those of the standard Teletype, although some CRT displays allow lower-case alphabet letters in addition. Unlike the Teletype, letters cannot be overprinted on the CRT to form new characters; previous images are lost to view. A cursor (usually a blinking point or line) indicates the position of the imaginary printer as it "types" each character. Output rates of 30 characters per second are available. A range of interesting graphics experiments is possible with the CRT display terminal.

The XY Plotter, another useful peripheral, consists of a frame holding a piece of paper (a couple of feet square) and supporting a penpoint so that it is free to move in two dimensions at right angles to each other in the plane of the paper. The dimensions are labeled X and Y. Each dimension is divided into a large number of "points," amounting to print positions just as on the Teletype; on the plotter, however, there are 50 to 100 positions per inch. Each possible position on the paper has both an X value and a Y value that uniquely locates the point in two dimensions. As the penpoint is moved from position to position, it leaves a continuous trace or line. With the plotter, continuous curves are possible, and colored inks may be used.

The computer may also be programmed to output through a high-speed line printer. I have encountered a number of interesting images produced on line printers. One, a striking representation of the Mona Lisa, was best viewed from a distance of about six feet. Line printers are rare in school settings at present, although they are commercially available.

The Turtle* is a small motor-driven robot which can be connected to the computer. It has a retractable pen so that it can leave traces as it moves over a flat drawing surface (floor, etc.) whose dimensions are limited only by the length of the cord connecting the Turtle and the computer. The movements are programmed. Turtles are not yet common in school settings, although several varieties are sold today.

*See Part I, p. 21.

The peripheral of the future art course may use a light pen to stimulate the CRT screen directly. Other advanced techniques for "sketching" or for animation may be commercially available in a decade or two.

With this range of graphics devices in mind, let us consider the roles of the computer in art instruction.

USES OF THE COMPUTER IN ART INSTRUCTION

Drill and Practice

The most obvious approach to art instruction at this level is in the area of knowledge of terms, facts, concepts—what we earlier termed the language of art, especially art history and criticism. With the programming languages now in use, a variety of objective test items (fill-in, multiple choice, etc.) can be constructed. In the context of secondary education, this role might be described as "Self-tester," provided the answers are given. A well-written art drill program would be in demand by students.

Tutorial

Simple languages are increasingly available for question-answer type programs with varying levels of complexity. A present-day teacher can (without too much trouble) learn to write fairly conversational branched tutorials.

Tutorials which include the special graphic abilities of the CRT or plotter, however, are rare. One such program displayed CRT pictures of the hand positions used by deaf persons to represent speech. The images were distinct although quite small. The student was asked to identify the alphabet letter corresponding to the hand position pictured, within a given time limit. It is admittedly a long way from such a development to a computer-directed drawing tutorial. A great many interesting art-knowledge tutorials could be designed with current equipment, given the right combination of teacher and programmer.

Problem Solving

The role of problem solver overlaps with that of data analyst in the field of art instruction and research, since data analysis is one of the main ways in which the computer solves problems in the field of art or art instruction. Consider such questions as:

(a) What clues or essential features do we use to identify human faces?
(b) How do you tell a real masterpiece from a masterly fake?

Such questions are now the subject of intense research by art experts working closely with computer experts. Their "solution" by students would increase tremendously their understanding of this growing field of scholarship.

Data Analyst

One of the main problems of the computer data analyst is formatting the output, after the computer has performed the arithmetic. Arranging the output so as to speed up the interpretation of the results is in some ways an artistic—or at least graphics—problem.

Turning this idea around, we might say that the data are a symbolic representation of the output image, be it a table of numbers, graph, geometric design, or portrait. Programs which "translate" data into images currently exist in two or more forms: (1) They are programs that produce the same image each time they are run. For example, a visitor to almost any educational computer facility will encounter one or more of the dozens of variations on Snoopy, the cartoon character. Many of these programs have been written by students in problem-solving mode. (2) Programs exist which allow the data to be changed from one run to another. Such programs are true translators. Many allow the data to be entered by the user from the terminal while the program is running.

Simulations and Games

Many of the program types discussed above may be simulations in addition to their primary application. For example, a translator program which produces a picture upon receiving a set of data from the terminal may be viewed as a simulation of a programmable graphics computer. (The idea of using one computer to simulate another computer may seem strange, but is is common computer practice.) Thus, many of the programs and problems I discussed previously could serve as bases for simulations. Let us look briefly at one interesting area not discussed above.

The production of animated films is an absorbing problem for artists and art students. Full-scale graphics computers capable of generating animated sequences are now too rare and expensive for use in the schools except on an experimental basis. Simple animation techniques, however, can be simulated on a common CRT display; the higher the character rate, the better the result. Programming animated sequences for such a medium could be tedious for an individual, less so for a team of students. The following type of programming task would also be a problem to solve: How does a television or similar device produce continuous, apparently moving images?

CONCLUSIONS

We have reviewed the use of the educational computer as an artistic or graphic medium to the possible exclusion of other applications in the art curriculum. There are two main reasons for this choice of emphasis. First, purely cognitive problems are relatively rare in art instruction. As indicated in the introduction, most of the learning is achieved through a combination of viewing or experiencing works of art directly, with the simultaneous (or nearly so) presentation of their symbolic interpretations. Presentation of verbal material is common in computer instruction. The production of images, of especial interest to art instruction, has received less attention. Second, computer-assisted or augmented art instruction must be relatively rare in today's schools. Since most concrete applications lie in the future, it seems appropriate to emphasize the computer technology of the near, and not-so-near, future. Computer graphics devices will be more available in the future than they are now; many are still in experimental stages in the laboratory or in selected school settings. Indeed, one could predict that computer art instruction will receive serious consideration from teachers and students only when interesting computer graphics devices become much more common than they are at the present time.

REFERENCES

Altair Design, developed by Ensor Holiday. New York: Pantheon, C. 1970.

Computer Graphics with an XY Plotter. Project Solo, Department of Physics, University of Pittsburgh.

Harmon, Leon D., "The Recognition of Faces." *Scientific American*, November 1973.

People's Computer Company (periodical). People's Computer Center, P.O. Box 310, Menlo Park, California.

Turner, Lawrence, Jr., *PILOT Reference Manual*. Hewlett-Packard Co. 1974 (in press).

Films of computer graphics, animation, etc., may be ordered from the catalogs of film distributors available to schools.

COMPUTERS IN THE BUSINESS EDUCATION CURRICULUM
by Eugene Muscat

INTRODUCTION

This article describes many current and projected uses of computers in supporting business education curriculum. Business education, as used in this discussion, includes all business subjects. Traditional courses such as bookkeeping, typing, and secretarial science receive as much attention as the more recent additions–keypunch operation, computer programming, and word processing.

For business education this is a time of change. Several current trends should greatly affect computer use:

1. *The emergence of individualized instruction and new teacher roles.* Computer-assisted instruction is an excellent addition to the learning mix when students are free to explore new concepts.
2. *Emphasis on competency-based curriculum and accountability.* The ability of the computer to analyze, diagnose, and prescribe should be part of any measurement or evaluation process.
3. *Development of regional occupational programs (county support).* New funding structures make it possible to develop "hands-on" programs using computer equipment.
4. *Technological advances.* Computer equipment has become less expensive and far more mobile. Where students formerly went to the computer, the computer is now coming to them by way of terminals linked with phone lines.

Because of such developments, opportunities for computer-assisted instruction, computer-managed instruction, and hands-on computer operation are greatly enhanced. The current trends are moving business educators to explore the capability of computer equipment and point the way toward greatly expanded use.

Increased usage logically raises questions about the computer as an instructional tool. To make equipment acquisition economically feasible, we need to consider the multiplicity of functions the computer can perform. Those interested in the computer as a teaching aid often have not experienced many of the computer's potential uses, so that projects can misrepresent costs and hamper utilization without an expanded view of the computer's instructional capabilities.

From the very start, a computer project in business education should include three uses:

1. As the object of instruction;
2. As the means of instruction;
3. As the manager of instruction.

In brief, the character and nature of the first two functions of the computer, as I observe them in business education, are as follows.

THE COMPUTER AS AN OBJECT OF INSTRUCTION

The entry of computer terminals can provide an excellent tool for occupational preparation in several new areas.

Key Entry

The existence of computer hardware in a school provides an opportunity to give hands-on training that should not be overlooked. Projections indicate that the largest growth area of clerical employment in the years to come will be in the area of key entry. Numerous p ograms can be developed (much like keypunch programs) to train students in the skill of key entry from terminals. Simple test routines that simulate the operation of an airline reservation system, an accounts receivable input system, and a ticketing reservation system provide excellent training for students who may enter the clerical field as data input clerks.

Simple programs have been developed that allow students to type in strings of information under various time constraints (VERIFY).[1] The computer will analyze the string according to predetermined criteria and individually tell the student his number of errors and speed of data input.

Word Processing

Because of the entry of various automatic typewriters and automatic transcribing equipment, computer terminals can expose students to the nature of such automated equipment as automatic typewriters, automatic typesetters, and test editing machines. Programs can be written using a standard Teletype and a digital computer to simulate the operation of these machines (MTST).[2] A student can be given first a standard letter to enter and edit on the computer terminal, then various inside address lists. This simulates a commercial computer-generated mailing project.

Computer Programming

The emergence of small standalone computer systems and programmable calculators has made online programming a valuable tool to a student entering the business field. He can readily learn to write programs in languages like BASIC by using on-line computer terminals.

Students should be encouraged to support regular classroom instruction in accounting, clerical record-keeping, sales, and so on by automating many of the clerical tasks. They can use keypunch machines, card readers, and batch processing, as well as on-line key entry. On-line capability developed in this way can be transferred to many business applications today.

1. Hewlett-Packard Users Group Library.
2. Hewlett-Packard Users Group Library.

Computer Operations

Where the CPU (central processing unit) is available for student use there is a variety of entry level skills that can and should be incorporated in an introduction to data processing curriculum. The use and management of various types of computer input media (cards, tapes, microfiche) should be included if possible. Experience in handling punch paper tape, for example, provides excellent training as a data entry clerk or tape librarian. The activities involved in maintaining a minicomputer system (mounting tapes, loading and unloading computer programs) should be shared by students who have access to actual computer equipment. It is the role playing in such school experiences that has greatest value. What justifies field trips and production work experiences is not so much the actual hands-on training with specific computer equipment, but the fact that students can feel the responsibility of performing these functions.

Some forms of computer technology may develop in schools where the electronics and/or industrial arts programs are highly specialized. Computer Teletype maintenance is an employment field that will take its place among typewriter repair and other business machine maintenance occupations. All in all, every opportunity to give the students actual experience should be exploited.

THE COMPUTER AS AN INSTRUCTIONAL TOOL

Teaching with computers is generally given the greatest exposure in discussing instructional computer use. Many approaches have been tested and evaluated; specific attempts in business education are detailed below.

Drill and Practice

Business education has traditionally dealt with the development of various skills: typing, machine computation, shorthand, cashier skills, and so on. The process requires extensive drill and practice. It is only logical, therefore, that the computer has been harnessed in many business applications as a drillmaster.

Typing. To develop keyboard accuracy students are given paragraphs to type, not for speed but for accuracy (TYPE).[3] The computer tallies the number of errors and indicates where specific key relationships are consistently violated.

Machine Calculation. Critical to machine calculation is placing decimals, estimating answers, and understanding basic mathematical functions. Computer programs which drill addition, subtraction, multiplication, and division are logical support tools here (DRILL). Drill and practice routines which test the student's ability to round numbers (ROUND) and place decimal points in answers (DEC1-10) can be used.

Shorthand. Shorthand requires a basic understanding of English skills. Programs which reinforce spelling and punctuation instruction are in wide use by secretarial-science students (SPELL). It might interest some teachers to know that even phonics and sound discrimination have been developed in computer-assisted instruction applications. Tape recorders and student response through keyboards have been designed to reinforce the student's understanding of various audible consonant and vowel sounds (PHONICS).

Cashier Skills. Sales transactions require extensive drill and practice. Before a student can be allowed the responsibility of dealing with cash, simulation is often desirable. Programs that drill students in how to make change (CASH) and compute sales tax (TAXIT) develop reflex skills that reduce extensive on-the-job training.

Accounting. What teacher has not longed for relief from the constant workbook and classroom drill required to establish absolute student understanding of the debit and credit principles? A computer programmed to analyze students' response can provide an unlimited number of ledger accounts from which a student can drill and practice and determine debit or credit balances. In general, wherever a specific skill needs to be developed—for example, multiplication tables and punctuation rules—the computer can provide a self-correcting method for drill and practice. The possibilities are limitless.

3. The programs TYPE, DRILL, ROUND, DEC1-10, SPELL, CASH, TAXIT, and TUTO1 described in this section are all in the Hewlett-Packard Users Group library.

Tutorial

In business education, numerous opportunities for tutorial instruction present themselves. Students in advanced classes in bookkeeping or office procedures, say, often are given materials and assignments to work on independently, with the teacher available only to answer questions when the student encounters difficulty. If we accept this role as the mere disseminator of materials, we can use the computer in a tutorial mode.

An excellent example is student interest in computer programming. In a school where only a limited number of students has the interest and/or capability to develop computer programming skill, the teacher may put the computer's tutorial capability to greatest use. Various tutorial programs teach a student BASIC programming (TUTO1). The lessons introduce the elements, provide examples, and encourage student experimentation—all at the computer terminal. Self-checking is provided by comparison of sample output with the result of the student-made program. This method has allowed many teachers with little background in programming to have students learn it with a minimum of student-teacher interaction.

In the same way such specialized teacher skills as experience with medical or legal terminology can be accomplished using a computer program as tutor and guide. Exposure of this type can be teacher-independent; the students can do it before or after school or during independent study. In the high school area this is particularly attractive; many colleges are developing tutorial programs in business education, and students with exceptional abilities can interact with tutorial curriculum to give them advanced placement course work.

In general, tutorial curriculum is the area of computer application that can best enrich business education students. In a school where a limited number of students would be capable of or interested in business law, for example, a tutorial business law unit could be prepared on a computer terminal that would be available for independent study. Similarly, tutorial units in completion of tax forms or highly specialized employment units can provide enrichment.

Problem Solving

Problem solving comes close to bridging the gap between the computer as a means of instruction and the computer as an object of instruction.

Using the calculating potential of a computer is a valuable experience for students who will find computers increasingly available to them as problem solving tools (i.e. as objects of instruction). This exposure has definite occupational value. Programmable calculators, automated cash registers, automated billing systems, ticket reservations systems, and automated graphics equipment will require experience in creating machine readable instructions.

As for curriculum support (i.e. means of instruction), the accounting student can tackle larger and more meaningful accounting problems if he has computer support. Comparing the effects of various methods of depreciation is greatly aided by the problem-solving capability of a digital computer. Instructors can effectively employ "canned" accounting problems that demonstrate business data processing in the accounting field.

Data Analysis

Opportunities for data analysis in the junior and senior high school business program are growing steadily. An obvious example is in teaching accounting. As mentioned above, comparisons of depreciation allowances can develop better student understanding of business decision making. In business law or consumer economics students can evaluate the cost of buying on credit. Energy-conscious students could evaluate the gas consumption of various types of autos in years to come.

Aside from the management games currently popular among high school students, data analysis techniques should be used to evaluate real world situations. The best business education project I can imagine in this area would be to collect and analyze an employment survey of entry level jobs in a student's community. The development of the questionnaire, computer input, and analysis could produce employment forecasts of immediate value and interest to the students and teachers of a high school area. This type of practical application can greatly enlarge the understanding of computer utility.

Simulations

Games often provide valuable instructional drill and practice. The game of hangman (HANGIT),[4] for example, is beneficial in developing word skills. Computer football, where the student is expected to maintain game statistics, is a valuable mathematics exercise (49ers). In general, the difference between educational simulations and games is often not the content but the attitude of the person participating.

This leads to the role of simulations in business education. Here, the teacher should control the curriculum content. Simulations of stock market activity can be administered in a controlled environment (STOCK). A class using a computer terminal can stress oral and written communications while "playing the stock market."

The value of a simulation (game) is generally beyond question. In business law numerous examples for student participation can be developed. A simple example of driver's education can demonstrate the penalty for improper judgment in highway driving (DRIVE).

In consumer protection examples, students are put in hypothetical buying situations and suffer or enjoy the consequences of their buying decisions. Developing student skills in preparing for and participating in job interviews is an active area: students are given hypothetical interview situations and asked to respond accordingly. The immediate critique of proper and improper responses can cue a student on the do's and don't's of job interviewing.

In general, simulations should be used to accomplish learning objectives that would be difficult or impossible in a normal classroom. An excellent example is taken from a bookkeeping simulation. Students are asked to submit worksheets containing a minor mathematical error. The computer is given this erroneous information and a multitude of computer produced end-of-month reports are created in a short time. Students trace the error and observe the consequences that a single mistake causes in a series of accounting reports. This is an impressive computer demonstration.

SUMMARY

The computer as the means of instruction has received a great deal of coverage in educational journals and conference tours. Thanks to the emergence of timesharing applications will continue to increase. All computer manufacturers currently have users' groups that make materials available. The increase of the

4. The programs HANGIT, 49ers, STOCK, and DRIVE are all in the Hewlett-Packard Users Group library.

number of time-sharing computer systems in secondary schools will cause a rapid increase of accessibility and use.

Unfortunately, computers are generally purchased not in order to tap instructional support capabilities, but for administrative purposes; still, they can be harnessed to instructional objectives. Regardless of the reason for purchase, the computer is an effective and supportive tool in business education. It is up to you as an innovative teacher to enlarge its use in your school.

REFERENCES

All of the computer programs referred to in this article have been contributed to the Hewlett-Packard User's Group library. Materials are available from:

Hewlett-Packard Basic Users Group Library
1100 Wolfe Road
Cupertino, California 95014

All other questions concerning computer support of business education programs can be directed to the students and staff of:

The E.D.P. Resource Center
San Francisco Unified School District
400 Mansell Street
San Francisco, California 94134

Educators are advised to contact the local chapters of the following organizations for additional help:

The Society of Data Educators
The Data Processing Management Association
The Association for Computing Machinery
The State Bureau of Business Education

COMPUTERS IN INSTRUCTION FOR THE DEAF AND HARD OF HEARING by Bruce D. Keepes

There is an unusual teacher's aide being tried out in a San Francisco classroom. He chatters, whirls, blinks, and in his fashion, treats the youngsters more humanely than many teachers. For instance, he never punishes. If a student makes a mistake, he does not remonstrate or give a failing grade. He simply says, "No . . . try again." He is Computer Tutor and he is so good at his job that some teachers are starting to worry that they could be replaced.

Stephen Cook, *San Francisco Chronicle*, 1973

INTRODUCTION

The teachers of acoustically handicapped youngsters in the Palo Alto Unified School District (PAUSD) are not worrying that they will be replaced. They support the use of a computer as a valuable aid in their instructional program. The how and why of this program may point directions not only for teachers of the acoustically handicapped but also for teachers of other handicapped youngsters and the nonhandicapped as well.

The computer-assisted instruction (CAI) project used by the PAUSD program had its origin in a Federal grant (Title IV-A) for the development, trial use, and evaluation of CAI materials for acoustically handicapped youngsters. Professor Patrick Suppes of Stanford University and Dr. Dean Brown of the Stanford Research Institute worked with the school district in planning the project.

The initial phase was to provide in-service training for teachers on the use of the computer and on writing lessons. Release time was provided so that the teachers could visit other classes and spend regular class time with professionals in the field. Later in the program teachers were given help in testing and evaluating their CAI lessons. The funding lasted for barely a year, but during that time the teachers wrote over 300 computer-assisted instruction lessons.

From this starting point the program has continued to develop into other subject areas and to expand the terminal time for each student. Existing lessons have been refined, new ones have been added, and the computer language in which teachers write the lessons has been extended.

Many people with little or no experience in communicating with handicapped children fail to realize that communication through speech is probably the least important facet of educating deaf children to function in a hearing world. The normal child acquires language without conscious effort; the ears serve as a bridge between spoken language and comprehension. The acoustically handicapped youngster, on the other hand, is deprived not only of that bridge, but of the concept of the function of language as a tool of communication. It has been estimated that during the first five years of life, the normal child of average intelligence acquires a vocabulary of approximately 2,500 words more than a deaf child will acquire.

The normal child probably uses these words after hearing them repeated only about six times. The average deaf child must be exposed to a word at least 60 times before it becomes a part of his vocabulary. Further, the complexity of our language—verb tenses, idiomatic expressions, varied definitions for a single word, unwritten grammatical rules that are acquired through hearing—magnify these differences at least 10 times. No teacher could remain sane if he had to repeat an idea or concept or process 60 times. Six times is difficult enough. The computer will repeat the same instruction 160 times if necessary without any emotion or strain as a tireless drillmaster.

One of the rationales for establishing the CAI program with deaf children was that they need more drill and tutorial experiences than the normal, hearing child. They also need more opportunities to learn in a situation in which their mistakes are subject to public display.

Beyond its use as a drillmaster, the computer also acts as a tutor for acoustically handicapped children. Communication between the teacher and the acoustically handicapped child takes much more time than with a hearing child; thus the process of learning a skill or concept is slow even when the subject matter itself is not a problem. It is particularly significant that in CAI the primary means of communication between the child and the computer is through a terminal, a visual device. This naturally speeds up the learning process. The computer provides lessons in a tutorial form. The child is presented with new information, new facts, and new concepts which are later used in the course of a lesson.

Further, the computer program can be individualized to fit a particular child's needs. If he needs only a certain part of a lesson, it can easily be copied and modified without destroying the original. Also, all lessons in the computer are available to all staff members at all grade levels at all times. Since the lessons do not carry grade levels, a high school student—if need be—can be assigned a lesson written by a second or third grade teacher with no stigma attached. Likewise, a very able student can use lessons written for children on a higher grade level, thus providing a challenge to his abilities.

THE ROLE OF THE COMPUTER IN INSTRUCTING THE HEARING HANDICAPPED

Drill and Practice

In the Palo Alto Unified School District, the major area in which the computer acts as drillmaster is in mathematics drill and practice. The programs were developed by Dr. Suppes, and the rights to use them were purchased by Hewlett-Packard, the manufacturer of the PAUSD computer. Hewlett-Packard, in turn, leases the lessons to us for $1 per year.

The lessons provide a series of exercises giving the child ample opportunity to practice skills previously taught through some other means. The rate of his progress through the materials depends upon the number of correct responses he has. While the curriculum is fixed and highly structured, the rate of progress is highly flexible and the children may work independently of the teacher.

The mathematics drill and practice materials, used mainly at the elementary level, are divided into six levels with 24 blocks in each. The teacher or proctor has control over the lesson the student is working on at any time, though if the teacher does not take any action, the child will be led systematically through all the blocks at each level. The computer maintains complete records on the progress of each student, and these can be recalled by the teacher when required.

Tutorial

In the tutorial role, the more than 350 lessons in grammar, geography, history, science, government, driver education, economics, and law prepared by PAUSD teachers present new information to the students. The student is presented with new concepts (or concepts previously introduced but not yet mastered) and given an opportunity to develop and demonstrate his mastery of them. If he is unable to do so, the computer lesson provides hints, re-presents the original stimulus material, or—if all else fails—gives the correct answers. Each lesson is maintained on a disc file in the computer so that all teachers have access to all lessons at any time.

The documentation for these lessons is contained in large binders provided for each teacher. Each lesson is documented on two sides of a single sheet of paper. The front side contains the name of the lesson, some information on what the lesson is about, special instructions regarding materials that might be needed by the students (for example, an art lesson requires that the student have crayons), and suggested grade levels. On the reverse side is a sample printout of a lesson taken by a student. The teacher who is unfamiliar with the lesson may quickly scan the sample page and develop a feel for the level of sophistication required.

Each acoustically handicapped student spends from 10 to 30 minutes per day working with the computer. The lessons, chosen by the teacher or teacher aide, are used in a variety of ways. Sometimes the material is first presented in class and then discussed and later reinforced through computer programs; sometimes new material is presented by the computer and later discussed in class. Many of the language lessons require the child to use an accompanying book, so that the tutorial aspect of the computer is presented in conjunction with a standard textbook.

TERMINALS AND LESSONS

All the terminals used in the acoustically handicapped program are hard copy. The computer printout is itself used in a variety of ways within the classroom. The students can bring printout sheets back to their class from the computer terminals, and these can then be used by the teacher for diagnosis and suggestions for further study. Sometimes a child takes a lesson and then is asked to explain it to other students; this gives him the opportunity to function in a role other than learner. The procedure provides the acoustically handicapped child with ample opportunity to practice his enunciation, pronunciation, sentence structuring, and other aspects of language in a meaningful situation.

The lessons written by the PAUSD teachers used a language called COPilot. This is similar to Pilot and many other CAI author languages such as coursewriter and IDF. The language is simple to use and most teachers learn to write lessons in a few hours. Most of the teachers in the acoustically handicapped program have written their own lessons and appreciate being able to do so. This skill enables them to view the computer as a tool for their use, not as a rival. The advantage of using an author language such as COPilot—lies not only in the ease with which programs can be written, but also in the way teachers can modify the programs for particular students. He can write his own materials and programs, or he can adapt existing materials. In some cases, students can also write programs, either for younger students or for themselves as a learning experience. COPilot has only 11 basic commands.

We have found that programs written by teachers use simple, concise language appropriate for children at their grade levels. The teachers tend to write in a lighthearted and lively manner, presenting concepts in a language readily understood by the children. (As an aside, many existing CAI programs, particularly those from commercial curriculum corporations, tend to be dull, drab, and filled with turgid prose.) Teachers writing their own lessons have an opportunity to observe the lessons as students take them, and so to modify and improve them.

SUMMARY AND CONCLUSIONS

Computer-assisted instruction for the acoustically handicapped has many advantages. First is the ease with which the children can acquire greatly needed visual repetition and drill on the many facts and skills which hearing children learn as they live. No teacher could find the time to do this effectively with each individual youngster. Although writing a computer-assisted instruction lesson is time consuming, once a lesson has been prepared it is always available whenever a need arises. Further, students can have an infinite amount of repetition if necessary.

Another time-saving feature is that most lessons are self-correcting. They call the student's attention to his correct answers and they either give him a chance to correct his mistakes or provide him with the correct answer.

The use of centrally-stored materials is highly advantageous. Materials prepared by one teacher can then be used by others; this is frequently not possible in a noncomputerized situation because teachers do not know what materials other teachers have prepared.

The computer allows the brighter child to progress as fast as he can by continually challenging him with new and more difficult concepts. This student need not spend as much time on a lesson as another child who is having more difficulty. For the child who has mastered the concepts, progress through the lesson is very rapid; for the child who is having difficulty, progress is very slow, allowing for a great deal of repetition.

Another advantage of computer lessons is removal of the peer-group competition that can make those with greater handicaps aware of their repeated failures.

The computer also increases teaching efficiency in that the child is saved hours of laboriously copying—often inaccurately—problems from a book before working them. The teacher is spared the monotonous hours of correcting the problems and discovering that in the time lapse children have repeated a number of errors many times, thus reinforcing them. With the computer, the error is not allowed to pass.

The teachers in PAUSD are extremely enthusiastic about the program, but have probably only scratched the surface. One teacher reported on a girl who, when she first started on the computer, even needed help in typing her name in order to sign onto a lesson, and then required almost continual assistance all through each lesson. Socially, the child would not speak to the teacher or to the other children and did not want the teacher to touch her. Her progress was really quite remarkable. By the end of two months, she was able to do lessons with almost no support from a teacher. During that period, her whole manner changed. She now sits and chats comfortably with the teacher and children, telling them about family plans, her lessons, and all sorts of other activities.

Parents are also enthusiastic about the computer programs and have on occasion suggested types of language concepts they felt would benefit the children; their suggestions have been incorporated into the lessons. Many others have volunteered to type COPilot lessons into the computer. The text is almost like English, and after typing a few, parents frequently find that they can write lessons themselves.

The use of the computer with acoustically handicapped children has not been without its problems. One of the greatest is that it is hard for a classroom teacher with no release time to write lessons. While the computer language is easy and the format of the lessons is fairly simple, writing a lesson is time-consuming and teachers can only be expected to produce a limited number of new programs each year. However, if each teacher produces even a small number of lessons a year, the overall effect in any one year is to increase significantly the number of lessons available in the common bank.

A second problem is scheduling the students for work at the terminals. Only a limited number of terminals are available and children have to be scheduled so that maximum use is made of the available time. This means that children are sometimes taken out of their regular classroom activities for work with the terminal, and so miss important classroom activities.

A third problem is system reliability. While computers are now developed to the point where they are extremely reliable (minicomputers can be expected to operate reliably with less than one failure per month), the modems, telephone lines, and Teletypes are not nearly as cooperative. It is extremely frustrating to both students and teachers—who are operating on a very tight schedule—to suddenly find that the computer is down.

A fourth problem is the cost. Fortunately, the cost of the computers is going down rapidly. The most economical size at present is one which will handle from 16 to 32 terminals, with an outright purchase price of less than $3,000 per port. If you add $1,000 or $1,500 per port for terminals (upper and lower case Teletypes with acoustical covers are very satisfactory), you have an initial expenditure of $4,000 to $4,500 per port. Maintenance on the system and terminals will cost approximately $25 per port per month.*

But given that we are gradually learning how to solve these problems and that the cost is decreasing, we consider that the program is a worthwhile investment of teacher and student time.†

REFERENCES

Brown, Dean, and Wiegand, Virginia. "Cost Effectiveness of Computer-Based Instruction of the Acoustically Handicapped: Prospectus for Research Project submitted to NIE," Stanford Research Institute; Menlo Park, California, 1974.

———, *A Catalog of COPilot Lessons*. Palo Alto, California: Palo Alto Unified School District, 1973.

———, *Computer Aided Instruction for the Deaf and Hard of Hearing: Final Report*. Palo Alto, California: Palo Alto Unified School District, 1971.

———, *Computer Aided Instruction for the Deaf and Hard of Hearing: Application for a Project Grant from Title VI, ESEA*. Palo Alto, California: Palo Alto Unified School District, 1970.

Coulson, John E., *Computer-Assisted Instruction and Its Potential for Individualizing Instruction*. Washington, D.C.: Academy for Educational Development, Inc., 1970.

———, *Educational Technology 1972–73*. Palo Alto, California: Palo Alto Unified School District, 1973.

Frye, Helen B., "Second Graders Like CAI." *Journal of Data Education*, 14, (1974).

Hansen, Duncan N., *Impact of CAI on Classroom Teachers*. Tallahassee, Florida: William L. Harvey, 1969.

*As of 1977, microcomputers offer the school computing capabilities for as little as $300 for one complete computer kit.

†The complete documentation of the COPilot lessons, COPilot language, and a magnetic recording of all lessons, the COPilot editor, and interpreter are available at a nominal cost from the Palo Alto Unified School District, Educational Technology Department, Palo Alto, California 94306.

Jackson, Ruth Montgomery, *Computer Assisted Instruction for the Deaf and Hard of Hearing.* Palo Alto, California: Palo Alto Unified School District, 1972.

Keepes, Bruce D., *COPilot Manual: An Author's Guide.* Palo Alto, California: Palo Alto Unified School District, 1974.

Keepes, Bruce D., Whitehouse, Len, and Grayson, Aram, *MINITRAN: A Mark Sense FORTRAN.* Palo Alto, California: Palo Alto Unified School District, 1974.

Stolurow, Lawrence M., *Some Factors in the Design of Systems for Computer-Assisted Instruction.* Cambridge: Harvard University, 1968.

COMPUTERS IN THE ELEMENTARY SCHOOL
by Donald Holznagel

INTRODUCTION

Viewed nationally, the elementary schools (including only grades one to six for purposes of this article) are currently making relatively little use of computers in comparison with secondary schools. On the other hand, some of the earliest large-scale efforts at investigating instructional computer applications were conducted in the elementary grades. While there are probably many reasons for both situations, one strong force in both is the limited view educators have had of what computers can do; a second is the limited view the computer industry has had of education and the task of teaching.

The initial and continuing major usage of computers in the elementary school is drill and practice, especially in mathematics. Certainly, drill is a tool of the teacher, and certainly computers can generate numbers quickly, make comparisons, and keep records efficiently. Just as certainly, however, drill is only one small aspect of the teaching task, and the aforementioned computer capabilities, along with others, make the computer useful in other ways.

An application area of great promise and of much current development work is problem solving, where researchers are investigating the ways in which children solve problems and ways in which the computer can enhance both the processes of problem solving and the learning of the processes. A third application—also of promise but of less current research activity in elementary schools—is simulation and gaming, for learning both subject matter and ways of modeling and problem solving.

As might be expected, there are differences of opinion on how the computer should be used in instruction. Some say the student should be highly active in the process, programming the computer, designing games, and controlling devices. It has even been suggested that the computer should not keep records about a student's terminal session, because that violates a certain trust the computer engenders in a student. Others say that any teaching task which is repetitious and well defined should be assigned to the computer, freeing the teacher for more humane tasks, keeping more accurate up-to-date information for decisions, and placing a student in a respondent role. Perhaps those are not mutually exclusive propositions, and aspects of both points of view can be represented in an elementary school to satisfy the diverse needs of a variety of students.

It is clear that elementary school educators will be making decisions about the nature of computer usage in their schools, and that to do so requires an understanding of what has been and can be done with computers. With that basis, they can begin to decide what *should* be done, consistent with the goals of children, parents, and teachers in the school.

THE ROLE OF THE COMPUTER IN ELEMENTARY INSTRUCTION

Drill and Practice

Elementary-school drill activities, as we have said, were one of the earliest instructional applications of computers, and that early work resulted in several major sets of commercially available programs. In mathematics, for example, programs are available which generate exercises randomly, accept the student response, check the answer, give reinforcement, and record student progress in a computer file. Such programs allow teachers to determine for students such factors as entry point in the curriculum, response time for a single item, and length of a terminal session. Rate of student progress and difficulty level of exercises are determined by the degree of student success on regular practice sets and periodic tests.

Speed and programmability are two computer capabilities of importance here. A computer can be programmed to compute random numbers within any desired limits. For example, to produce subtraction exercises one could ask for random pairs of whole numbers, the first being between 20 and 50 and the second being between 0 and 9. Thus, two students at the same point could receive similar, but not identical, sets of exercises. The high speed of computation allows the computer to generate exercises for many students and to carry out the required record-keeping in a manner which appears to be simultaneous for all the students and with almost immediate computer responses to every student's input.

Most programs allow teachers to intervene and enroll, delete, or change the placement of students in the curriculum. A teacher may get many reports on the status of individuals or classes of pupils based on the records of student sessions kept by the program. Although the number and type of reports vary, most programs include general information reports (names, I.D. numbers, parameters), performance reports (scores on sessions and tests), and daily activity reports (who has had a session, location in curriculum, unusual success or failure, encountering new concepts).

Three current major commercial sources of mathematics drill and practice are Computer Curriculum Corporation (*Mathematics Strands*, gr. 1–6), Houghton-Mifflin Co. (*Individualized Computational Skills Program—Computer Version*, for gr. 3 and up), and Hewlett-Packard Corp. (*HP Drill and Practice in Mathematics*, for gr. 1–6). In addition to mathematics, Computer Curriculum Corporation (CCC) provides programs in reading for grades 4–6 and in language arts for grades 3–6. The CCC products are the results of developmental efforts directed by Dr. Patrick Suppes at Stanford University. Other centers of development for elementary school programs and materials have been Project REFLECT in Montgomery County, Maryland, and the Flint, Michigan, Public Schools.

Large-scale studies of the effectiveness of computerized drill and practice have been carried out in McComb, Mississippi; New York City; Palo Alto, California; and the suburban Twin Cities area of Minnesota. Current usage is widespread, with heavy use in Chicago, Philadelphia, and Los Angeles. The Chicago school system has centered its elementary school computer use on drill and currently is supporting approximately 500 terminals.

Most large-scale studies have been done with large numbers of terminals, supported by grant funds. Since that is not a typical situation in the average school district, the Twin Cities area study attempted to determine the effectiveness of drill and practice in more common situations, assuming only one terminal per school and a variety of school organizational structures and philosophies. It was conducted in grades 3 and 4 of nine suburban and rural schools of TIES, a cooperative of 34 Minnesota school districts. The results indicated that drill and practice is especially effective with low-ability fourth graders who have daily sessions of 5–7 minutes and that these students showed greater progress than similar students not having computerized drill. Studies in the other situations also indicate more rapid and efficient mastery of skills when students are given daily computerized sessions of 5 to 10 minutes' duration. The other advantages claimed for computerized drill over manual methods are the more efficient and complete record-keeping and reporting, the relief of teacher effort in generating voluminous nonidentical exercise sets, and the highly individualized nature of drill sessions. Offsetting these, however, is the teacher time required to study and make use of the reports and to manage the logistics of terminal usage when a limited number of terminals is available. Furthermore, the cost of terminals, computer time, and programs could be considered prohibitive.

Some educators, feeling that the most valuable computer application is the generation of random nonidentical sets of exercises, have attempted to focus on programs to be used specifically to generate such sets in clear, copyable form as masters for duplication. Such programs keep no records and do not provide for student interaction at the terminal. The teacher can prescribe the type of exercise set by entering the appropriate number limits, number of problems, and problem types on the terminal, and will receive as many different versions as desired. A complete answer key for each master can also be generated at the same time. Such programs have the advantages of not typing the student's drill session to terminal availability or school building open hours, and of freeing the terminal for other applications. A major development effort in this regard is under way in the Minneapolis Public Schools, funded by the Minnesota Council on Quality Education.

It is important to keep in mind that, in the foregoing discussion of drill, the computer is viewed not as teaching but rather as providing an improved way of presenting a classic supplement to the teaching-learning process. There are those who do not agree with this use of computers. One view in opposition is that the proper use for the concept of drill programs is for students themselves to develop and write the programs, because it forces them to define processes and consider all possible mistakes.

Tutorial

Tutorial mode implies a dialogue between a computer program and a student, where the program is the tutor. Such programs are developed or authored by teachers or others having an understanding of the procedures of tutoring and a knowledge of the topic to be taught. Several computer languages invented specifically for this purpose are available on various computers and are intended to provide an easier medium for nonprogrammers to use in designing drill as well as tutorial lessons.

Tutorial lessons vary from tightly controlled programmed instruction to wide-ranging dialogues where many paths may be followed depending on the student response. The advantage of using a computer rather than printed material in this application is that the computer can be programmed, through keyword searches and other methods, to interpret student statements of any kind and respond in a wide variety of ways. It could be caused to issue the same response to two student statements having different wordings but the same meaning, or to issue different responses to two identical student statements.

Two prominent centers of tutorial material development for elementary schools are Project REFLECT in Montgomery County, Maryland, and the PLATO project at the University of Illinois. Products from both places are mostly in mathematics but also include reading, language arts, and other topics. Both projects make use of special terminals. The PLATO Terminal is of particular interest for its technology and usefulness and great long-term implications. Heavily funded by the National Science Foundation, the PLATO project could generate a large amount of lesson material in the near future.

A problem of increasing concern is that, although the newer author languages are easier to use and teachers can author lessons without also becoming computer programmers, the same languages are not often available on computers of different manufacturers. Hence we are facing the prospect of increasing availability of lesson material but with limited ability to share the results with colleagues on a widespread basis. Some would argue, however, that in order to properly interpret responses, one needs to know the way the student uses language, which can be different in two groups of children, hence limiting the desirability of sharing lesson material.

Problem Solving

As a category of applications, the problem-solving label covers a variety of activities wherein the involvement of the child ranges from very passive to very intensive.

A common activity is the use of a prepared program called from a computer library to compute results for data supplied by the student. The goal is not to understand the solution algorithm—it is assumed that the child already understands the process and that his repetition of the calculation is not beneficial or desirable—but rather to use the results to further understanding of a topic of study. A common example which might have use in elementary school is a program to find the average of a set of numbers, perhaps gathered in a science experiment or a survey.

Most current emphasis in usage and investigation is in student-written programs to solve problems related to such subjects as math and science and in the concomitant learning of procedures of problem solving. This activity requires the student to learn a programming language. Most commonly available to elementary age children is BASIC; some usage of it can be observed in many elementary schools. The Burnsville, Minnesota, Public Schools, for example, have developed a curriculum guide for use in their elementary school computer which specifies teaching some elements of BASIC in the fifth grade and using BASIC for general problem solving in the sixth grade. Other school districts around the nation are trying such activities, although perhaps not many to the same degree as a mandatory part of the curriculum.

Two other computer languages, PILOT and LOGO, are the focus of current investigation and development efforts with special emphasis on elementary age children. The center of PILOT activity is the Stanford Research Institute. Versions of the language are working on several different kinds of computer systems and usage is spreading. PILOT has been used

BASIC

```
10 PRINT "HOW OLD ARE YOU"
20 INPUT A
30 IF A > 10 THEN 60
40 PRINT "YOU'RE A BABY YET"
50 GO TO 70
60 PRINT "GEE, YOU'RE ANCIENT"
70 END
```

LOGO

```
10 PRINT "HOW OLD ARE YOU"
20 TYPE IN "ANS"
30 TEST: ANS > 10
40 IF TRUE PRINT "GEE, YOU'RE
                      ANCIENT"
50 IF FALSE PRINT "YOU'RE A BABY
                          YET"
END
```

Figure 4-4. A sample program in BASIC and LOGO.

extensively by elementary children in the San Francisco Bay area.

The LOGO language was originally developed at Bolt, Berauck, and Newman and Massachusetts Institute of Technology. It has been used as a tool for investigating the nature of problem solving and for teaching problem-solving methods by researchers at Syracuse University in addition to the original development groups, and some work has been done at the University of Pittsburgh and in Montreal.

The following statements illustrate some of the beliefs of those who say that student programming enhances problem solving and the learning of problem-solving procedures:

1. Children learn by doing things and by thinking about what they do.
2. Programming helps students think about the process of solving problems by forcing them to define algorithms.
3. The act of computer programming facilitates the acquisition of rigorous thinking.
4. Programming helps develop systematic thought processes in approaching problems.
5. Programming helps students learn certain mathematics concepts.
6. Programming provides a laboratory framework for exploring mathematics.

If those principles are correct, we need programming languages which are powerful yet simple to learn and use, so that they can be taught to young children. BASIC, PILOT, and LOGO are efforts in that direction. LOGO, the latest of these and designed specifically with children in mind, has been the focus of the most up-to-date research in teaching problem solving, having been taught to groups of students in every grade level from 2 through 7. In one such research effort, the Syracuse project examined four questions:

1. What content areas can be supplemented or replaced by LOGO materials?
2. What contribution can LOGO make as a separate curriculum area?
3. Do children become more explicit and logical in their thinking as a result of working with LOGO?
4. Does LOGO develop inquiring attitudes and self-sufficiency?

Results on question 1 suggest that children gain much mathematical experience in a concrete fashion not ordinarily available; some even contend that LOGO could form the framework for the mathematics curriculum. Regarding question 2, it was evident that children do gain skills in planning and solving problems, and perhaps problem solving as a topic should have a place in the curriculum. Evidence is inconclusive on the last two questions.

The example in Figure 4-4 illustrates a simple student program in both BASIC and LOGO (intended only as a sample—not a definitive comparison).

Simulation and Games

This is an area of great potential, although not much developmental activity has taken place. The most extensive early work was done by Richard L. Wing and others in a research project in Westchester

County, New York, during 1965-1967. Three computer-based economics games for the sixth-grade level were developed and tested on a small scale with a group of sixth-grade students. The testing did not yield any conclusive evidence on the effectiveness of the games but showed that it was possible to use computer simulation in the elementary school, that the experience was enjoyable, and that the same material was taught in less time than by noncomputer methods. The three programs were the Sumerian Game, the Sierra Leone Development Game, and the Free Enterprise Game. In each, the students play a decision-making role (e.g. a city-state ruler in the Sumerian Game) and are expected to make decisions about the allocation of resources. As they continue the simulations, they should begin to note the consequences of their decisions and hence learn something about the nature of the economic system in which they are operating.

Learning about the nature of the system is one objective of using simulation as a teaching tool. If the model is close to reality, one could expect students to learn something about reality by engaging in the simulation. At the same time, the student is experiencing a computer application which is widespread in government and industry. For example, computer simulations of river systems are used by engineers to see the effect of a decision on dam construction, dredging, or other change without having to suffer the consequences. A classroom parallel might be a simulation of population growth in an animal species being studied. Students could manipulate the allowed hunting harvest and note the effects on the size of the herd. They do not need an actual herd on hand, and their decisions effect no actual extinction, overpopulation, or other disaster to the species.

But simulation activities for classrooms have been available for a long time, so why use a computer? One reason is that because the student is not required to carry out calculations, the model can be complex and therefore more realistic. Another reason is that the computer can generate random numbers quickly and hence introduce random factors (drought, disease) in a way which appears more realistic because no dice-rolling or card-drawing is necessary as in a manual simulation.

Generally, computer simulation could allow elementary students to experience situations not otherwise possible in a classroom and to learn about one method of problem solving with computers used in many walks of life. The lack of developmental effort, however, means that not enough programs are currently available to make a significant impact in elementary education. The teacher is left to make a professional judgment on the value of a given game or simulation for a specific class or situation, and would probably find as many colleagues in agreement as in disagreement.

Computer Literacy

The term "computer literacy" has come to be used to describe the knowledge most people need to function in, and understand the computerized aspects of, society. It is possible for elementary students to learn many such concepts. In fact, if such knowledge is expected of all citizens, elementary teachers could assume responsibility for providing it, since the elementary school is almost the only place we can still ensure a common experience for all children.

Using a computer in any of the modes discussed is teaching something about computers and their usefulness. If a computer is used only for drill, a student has a very narrow vision of its power and value. Therefore, one step toward giving students a broad view would be to make use of many application modes. Another step is directly teaching computer concepts. The Burnsville School District mentioned on page 00 begins its elementary computer curriculum with some instruction to all fourth graders in computer concepts. A variety of demonstrations can be designed and some actual student programming can be done; such activities have been successful in many schools.

SUMMARY AND CONCLUSIONS

The computer can be viewed by teachers as a sophisticated, important medium for expression, but—like other media—it must be used where it offers an instructional advantage. If drill or simulation is inappropriate in a given situation, doing them on a computer in millionths of a second is no more appropriate, however spectacular. Educators need to avoid being astounded by what computers can be made to do and become concerned about what they *should* be made to do.

Our application of computers to instruction has been conditioned by our view of the process of education. The computer can be used to do, perhaps more efficiently, some of the things teachers have always done, and it can be used to do things teachers have not been able to do. Whether either is worth the cost depends on how highly one values those practices or possible changes. The question of what constitutes good instruction has never been answered by one person for all others. It is incumbent on teachers to learn about computers and about the processes of education, and to apply the former to those aspects of the latter which receive their priorities.

As a framework for summarizing applications, Figure 4-5 is offered. One can easily extend it, but it covers the current need. The C indicates an area of current or past effort and the F indicates areas which appear to offer the most fruitful ground for future efforts at elementary school computer usage (not excluding further work in C areas).

MODES	*Math*	*Reading*	*Lang. Arts*	*Soc. Stud.*	*Sci.*
Drill and Practice	C	C	C		
Tutorial	C	F	F	F	F
Simulation	F			C	F
Problem Solving	C			F	F
Computer Literacy	C			F	F

Figure 4-5. Current (C) and future (F) curricular areas for computer applications in elementary schools.

While the goal in this article is to discuss the computer as an instructional tool, a computer so used can also be used effectively to assist the management of student experience and activities; in fact, such usage is receiving much attention. This is another story worth pursuing.

Of course, there are problems of logistics, physical limitations, and technology which affect any attempt at computer use. The cheapest available printing terminal is also the noisiest and most cumbersome. Telephone transmission is subject to the vagaries of extraneous noise and electrical disturbances. Inconvenient locations of power outlets and telephones impose classroom control problems (see the Martin reference for a more complete delineation). One or more of these factors is sufficient to invalidate computer usage in a given situation, in the same manner as the 16 mm film which fails to arrive on the day it would be most effective. While contingencies must be planned for and problems considered, the reliability and quality of technology appears to be moving forward rapidly, and we can expect a fairly rapid convergence of decreasing cost and increasing reliability with increasing readiness of teachers to explore computer applications.

The article attempts to introduce the uninformed reader to the variety of current computer applications in elementary school usage, to summarize some efforts in the most prominent applications areas, and to indicate areas of likely future effort. It is too brief to be exhaustive in sources and reference and in the reporting of what is known. The reader is urged to delve more deeply in areas of special interest; the following references are a good starting point.

REFERENCES

Brown, Stephen I., et al., *Final Report: June, 1973.* SU/LOGO Project, SURC-EPRC. Syracuse, N.Y.: Syracuse University, (Excellent material on problem solving in elementary school)

Curriculum Package Descriptions and Sample Lessons. Palo Alto, California: Computer Curriculum Corporation.

Feurzeig, W., and Lukas, G., "LOGO—A Programming Language for Teaching Mathematics," *Educational Technology*, March 1972. (A general discussion with examples.)

Introduction to HP Drill and Practice. Hewlett-Packard Corp.

Martin, Gerald R., *The 1972-73 Drill and Practice Study.* Minnesota School Districts Data Processing Joint Board (TIES), 1973. (Available from TIES, 1925 County Rd. B2, Roseville, Minnesota 55113.)

Papert, S., and Solomon, C., "Twenty Things To Do With A Computer." *Educational Technology*, April 1972. (Some fun things done by elementary students.)

Project REFLECT: Final Report. CAI Project, Einstein High School.

Rubin, Sylvan, "A Simple Instructional Language." *Computer Decisions*, November 1973. (A brief discussion of PILOT and its predecessors.)

Saturday Review of Education, May 1973. See pp. 48-50 for Suppes, Albrecht, and Papert viewpoints on computers in education.

Shaw, Schaefer, and Hiehle, *Teacher's Manual: Individualized Computational Skills Program—Computer Version*. New York: Houghton-Mifflin Co., 1973.

Wallace, Scott, "Bridging the Gap Between Teacher and Programmer." *Computer Decisions*, June 1972.

Wing, Richard L., et al., *The Production and Evaluation of Three Computer Based Economics Games for the Sixth Grade*. Yorktown Heights, N.Y.: BOCES First Supervisory District, June 1967.

COMPUTERS IN LANGUAGE ARTS by David H. Ahl

INTRODUCTION

The computer has been the subject of novels for years and has had a profound influence on both serious and fictional literature of the past 20 years, but it has barely infiltrated into the English curriculum. In an embryonic way, however, the computer is beginning to influence the teaching of English, largely because of its power to motivate.

Unlike science, in which our bounding technology has forced realignment of the curricula every five years or so, or mathematics, which has seen the invasion of the New Math and computer programming, changes to the English (or Language Arts) curricula have been of an evolutionary nature. Some of the traditionally rigid rules of grammar are being de-emphasized during the earlier years to encourage students to write more fluidly and to approach original composition with less apprehension. Nevertheless, grammar, spelling, word definition, sentence construction, and the like must still be taught at some point in order to assure a reasonable level of literacy and communication skill.

As it has become more widely recognized that performance in school as well as in a career is closely tied to communications ability and reading comprehension, there has been increasing emphasis on making the reading portion of the curriculum both relevant and fun. One approach is teaching on three levels—facts, concepts, and values—instead of facts alone. For example, on the facts level a student may be required to memorize all or part of a poem. He must know the author and his background, when the poem was written, its rhythmic pattern, and so on. On the concepts level, however, the student interprets the meaning of the poem and perhaps looks into how the heritage of the writer might have motivated him to write such a piece. One step further, on the values level, the student might be asked to relate the meaning of the poem to his own everyday life or to write a piece expressing the same message in contemporary terms.

The computer holds an exciting spectrum of possibilities in the area of Language Arts. Today, the seed has barely germinated. Just as the slide rule, calculator, and finally the computer have become welcome tools to mathematicians and scientists, there is no reason to think the computer should not become an extension of the hand or intellect of a writer. Poetry, some of it quite moving and readable, has been produced with the aid of a computer. The tool will not make an illiterate student into a great author; however, it may well give the reticent student the confidence he needs to tackle a writing assignment on his own.

Perhaps most promising in Language Arts is the use of computer games, to make reading, grammar, words, and other facets of English more fun. The computer, in other words, can motivate students so that they really want to read and write. For example, there are games that assist the student to clarify word definitions or express thoughts in a clear, concise manner. Other games help students take the first steps in original writing by helping them write a poem or humorous story.

Today in Language Arts, the computer is most often used in a drill-and-practice role, drilling the student on spelling, word meanings, plurals, rules of grammar, and the like; it is also used as a tutor for the same basic concepts. Occasionally, the computer plays a data-analysis role: for example, it can analyze the frequency of word usage in a novel or tabulate line and word length to determine readability.

USES OF THE COMPUTER IN LANGUAGE ARTS

Drill and Practice

This is the most common role of the computer in Language Arts. By giving the student drill and practice on concepts that have been previously presented, it relieves the teacher to devote individual attention where it is needed. With the computer, the drill and practice is self-paced and can be presented until the student achieves mastery of the necessary basic English skills.

Here are two examples from drill-and-practice programs:

Example A

```
IF THE FOLLOWING WORDS ARE CORRECTLY SPELLED,
RESPOND Y ; IF INCORRECT, TYPE THE CORRECT SPELLING.

EMBARASS
?Y
WRONG.  CORRECT SPELLING IS EMBARRASS

GAUGE
?Y
CORRECT!

SEIZE
?SIEZE
WRONG.  THE WORD WAS CORRECT
```

Example B

```
HERE IS A SENTENCE:

  GORILLAS OFTEN SLEEP IN TREES.

WHAT IS THE SUBJECT OF THE SENTENCE?
?GORILLAS
RIGHT
WHAT IS THE VERB?
?OFTEN
NO, TRY AGAIN
?SLEEP
CORRECT
```

Figure 4-6. Sample interactions with two drills: (A) spelling and (B) parts of speech.

Tutorial

Figure 4-7 contains two examples of tutorial programs suitable for use in Language Arts. The first is a traditional tutorial; the second combines reading comprehension with word definition.

Data Analysis

As mentioned in the introduction, the computer can analyze the frequency of various words or constructions in a written piece, or perform other statistical analyses on the content of textual material. For the most part these applications have been limited to research environments and have not reached the point of classroom application. For example, the writing of Bertrand Russell in *Mysticism and Logic* is much heavier than that in Clarence Day's *Life With Father*. While what is heavy or light reading is normally a subjective judgment on the part of the reader, perhaps a computer rhythmical analysis of phrases can shed some light on the terms in a quantitative manner.

Games

Many students are apprehensive about writing original work; even more are "turned off" to English and Language Arts. By use of the computer, Language Arts can be made more fun. Original composition can be introduced in a gradual, nonthreatening way and students can learn to express themselves without fear of ridicule or embarrassment.

Excerpts and exercises from four computer games applicable to Language Arts are presented on pages 136–138. BUZZWORD, an entertaining game, introduces new words to students and motivates them to find out what they mean. MADLIB also presents word definitions and synonyms in an enjoyable game.

The poetry programs go one step beyond MADLIB and help the student with a poem which, in a sense, is completely original. The computer uses phrases provided by the student but arranges them in a random manner—a simple technique but extremely motivating.

The ANIMAL game is enjoyable and addictive. It also teaches efficiency of expression in a most unexpected manner. The computer program simply asks the student to teach it new animals (it starts knowing only a bird and fish) by giving it a question to distinguish each new animal from one it already knows. Easy at first, but wait until the student has to state a question which can be answered yes or no that will distinguish a panther from a leopard. Not only does the student learn about animals (which is secondary) but he learns to describe differences between similar things in a clear, concise manner—and has fun doing it!

Example A

```
TWO OR MORE WORDS, USUALLY CONNECTED BY A CONJUNCTION,
ARE FREQUENTLY SUBJECTS OF ONE VERB.

USUALLY THE VERB WITH A COMPOUND SUBJECT IS PLURAL:
PAT AND STAN HAVE THE SAME COLOR EYES.

MUMPS AND CHICKEN POX (HAVE, HAS) BROKEN OUT IN CAMP.
?HAVE
CORRECT

BOTH THE GRAMMAR AND STYLE OF ADVERTISING (VARY, VARIES)
CONSIDERABLY FROM FORMAL ENGLISH.
?VARIES
NO, TRY AGAIN.  THE SUBJECT IS 'GRAMMAR AND STYLE'
?VARY
CORRECT
```

Example B

```
          BRAD HAS A STRONG COMMITMENT TO REACHING HIS GOALS.
          OCCASIONALLY, PROBABLY UNINTENTIONALLY, HE STEPPED
          ON BOTH TOM AND DAVE WHILE TRYING TO REACH HIS
          OBJECTIVE.

IN THE PARAGRAPH ABOVE, BRAD IS PROBABLY PLAYING
IN A FOOTBALL GAME (Y OR N).
?N
CORRECT.  THE GOALS SPOKEN OF ARE BROADER THAN THAT.
IN THE PARAGRAPH, COMMITMENT MEANS:  A)  REFERRING A MATTER
TO ANOTHER, B)  FAITH, C)  PERSONAL PLEDGE, D)  ACT OF
PERFORMING.
?B
FAITH IS CLOSE IN MEANING, BUT AS COMMITMENT IS USED HERE,
IT IMPLIES MORE THAN FAITH.
?C
YES.  BRAD HAS PROMISED HIMSELF THAT HE WILL DO HIS BEST TO
REACH HIS GOALS.
UNINTENTIONALLY MEANS:  A)  WITHOUT STRAINING, B)  WITHOUT
PLANNING, C)  NOT BEING PRESENT, D)  GETTING MIXED UP
?C
NO.  YOU MAY BE CONFUSING 'ATTEND' WITH 'INTEND'
INTEND AND INTENT HAS TO DO WITH ONE'S PURPOSE AND PLANS.
?B
CORRECT.  BRAD DID NOT INTEND OR PLAN TO STEP ON TOM AND DAVE.
```

Figure 4-7. Sample interactions with two tutorials—
(A) subject verb agreement and (B) comprehension.

PERCENTAGE DISTRIBUTIONS OF RHYTHMIC PATTERNS		
PATTERN	RUSSELL	DAY
RISING FEET	48%	41%
FALLING FEET	13	21
CREST	26	24
TROUGH	0	11
LEVEL	13	3
TOTAL	100	100

Figure 4–8. Sample data analysis run in language arts.

The Game of BUZZWORD

Everyone has heard of buzzwords. In these days of specialization, every profession has his share. Buzzwords generally describe something new for which there is no standard dictionary definition. They frequently become overused so that people feel obligated to use them in speeches or reports simply because it is expected.

EXERCISE 1.

Using the "educator-speak" buzzword generator below, take a random word from Column 1, one from Column 2, and one from Column 3 and create a buzzword phrase. Do this 10 times. How many of the individual words do you know the meaning of? Look up the words you don't know. How many of the 3-word phrases make any sense and how many are sheer nonsense?

1	*2*	*3*
ability	learning	grouping
basal evaluative	evaluative	modification
behavioral	objective	accountability
child-centered	cognitive	process
differentiated	enrichment	core curriculum
flexible	scheduling	algorithm
discovery	humanistic	performance
heterogeneous	integrated	reinforcement
homogeneous	non-graded	open classroom
manipulative	training	resource
modular	vertical age	structure
Tavistock	motivational	facility
individualized	creative	environment

EXERCISE 2.

Run the program BUZZWD, which produces 3-word phrases from high technology industry, or SPEECH, which produces 3-word phrases in the language of educators. Run the program you have selected 10 times. Again, how many phrases make sense and how many are nonsense?

OPTIONAL PROJECT.

For one of the professions listed below, find 30 buzzwords and write a buzzword generator of your own.

Medicine	Civil Engineering
Law	Social Work
Finance	Psychiatry
Women's Wear	Entertainment
Transportation	Agriculture

The Game of MADLIB

Madlibs, a creation of Roger Price, was designed as a hilarious party game. The leader has a sheet of paper like the one below, containing a short story with certain words missing. Before reading the story he asks each person in the room to supply one word—the first person may be asked for an adjective, the second for a noun, etc.—until all the blanks are filled in. The leader then reads the story to the group in a hearty, booming voice. Figure 4–9 is a sample madlib. With MADLIB, we can exercise real creativity with synonyms. Consider: instead of "big," we could say "huge," "tremendous," "enormous," "bulbous," "bulging," "massive," or "boundless."

EXERCISE 1.

For each of the following words, write 10 of the most ludicrous synonyms you can think of. Try to do it by yourself, or use a dictionary or Roget's *Thesaurus*. Compare your word lists with those of other class members.

NEWSPAPER ARTICLE

Mrs. Fifi Vanderbold, the ______ (adj.) and ______ (adj.) heiress has filed for divorce from her husband, Percy Vanderbold, the former ______ (adj.) ______ (noun) of Harvard, class of '38, now in the ______ (noun) business. Mrs. Vanderbold claimed that her husband had ______ (adverb) given her a ______ (noun) in the eye and had kicked her twice in the ______ (noun) and the ______ (noun). Mrs. Vanderbold, when asked to comment, said "______ (exclamation)! This is a ______ (adj.) lie. I only pinched her in the ______ (noun)."

Figure 4-9. Sample Madlib story sheet.

small	rough
old	ugly
witty	brain

EXERCISE 2.

Play MADLIB or MADLB2 on the computer. Set the tone of the finished story by using all ugly sounding words or all political words or all buzzwords from one profession (from BUZZWORD Exercises). Compare your stories with those of other class members.

Writing Poetry by Computer

Poetry does not necessarily need rhymes, regular meter, or a particular structure; it can be a combination of words that tell a story, convey an impression, or express emotions and feelings.

EXERCISE 1.

Run the computer program POET for a page or two of output. This is heavily random except that phrases from four groups generally follow in order. The phrases, for the most part, are similar to those that might be used by Edgar Allen Poe. Some of the stanzas will seem to make sense. Overall the impression is one of evil darkness and impending doom.

```
O.K.  WE'RE GOING TO WRITE SOME MAD-LIB STORIES
TOGETHER.  YOU SUPPLY THE WORDS, I SUPPLY THE PLOT.

FIRST I NEED 11 ADJECTIVES.  ADJECTIVES MODIFY
A NOUN, E.G., RED, DIRTY, LARGE, ETC.
?RED
?DIRTY
?LARGE
?UGLY
?CROOKED
?MESSY
?WHIRLING
?CONFUSED
?MOLDY
?FREEZING
?HAIRY

NOW I NEED 5 ADVERBS.  THEY MODIFY VERBS
AND END IN "LY" (SLOWLY, GREEDILY, ETC.)
?SLOWLY
?GREEDILY
?SLOPPILY
                DAILY GRUNT CLASSIFIEDS

FOR SALE:  1957 SEDAN.  THIS RED CAR IS IN A DIRTY CONDITION.
IT WAS FORMERLY OWNED BY A LARGE SCHOOL TEACHER WHO ALWAYS
DROVE IT SLOWLY.  THERE IS A UGLY INSECT IN THE BACK SEAT.  IT
HAS A CHROME TELETYPE ON THE HOOD, A CROOKED PAINT JOB, MESSY
TIRES AND THE BACK OPENS INTO A WHIRLING PENCIL.  WILL CONSIDER
TAKING A SLIGHTLY USED MARTINI IN TRADE.

LOST:  IN THE VICINITY OF MAYNARD DUMP, A CONFUSED FRENCH
POODLE WITH MOLDY HAIR AND A FREEZING TAIL.  IT ANSWERS TO
THE NAME OF SALLY.  WHEN LAST SEEN IT WAS CARRYING A GARBAGE
CAN IN ITS MOUTH.  A HAIRY REWARD IS OFFERED.
```

Figure 4-10. Sample run of MADLIB.

EXERCISE 2.

Divide the class into small groups. Have each group select one of the groups of words below and use them in the computer program BARD. This program is the same as POET except it allows you to input your phrases instead of using those already in the computer.

NATURE PHRASES	*INDUSTRY PHRASES*
Carpet of ferns	Harnessing energy
Morning dew	Industrial might
Tang of dawn	Steel, oil, timber, nylon
Swaying pines	Furnaces roaring
The song of nature	Around the clock
Entrances me	Mining, casting, refining
Soothing me	Computer tapes whir
Rustling leaves	Hammers pounding
Gently caresses	Throbbing, pulsating
Radiates calm	Rubber, zinc, glass
Mighty oaks	(You write these 5 phrases)
Grace and beauty	
Silently singing	
Nature speaking	
Captures my senses	

Untouched, unspoiled	Machines and people
Shades of green	Improving life
Tranquility	Conquering poverty
. . . Evermore	Energy . . .
So peaceful . . .	Technology . . .

EXERCISE 3.

Choose a subject and make up 20 phrases about it. Use the phrases in the program BARD.

EXERCISE 4.

Read the computer-generated poetry below. Do you like it more or less than poetry written by humans? Why?

Since the computer was programmed by a human, maybe we should regard the computer as an extension of the hand or the intellect of the writer. In this case the poetry is still really a product of the writer. Do you agree? Why or why not?

OPTIONAL PROJECT 1.

Write a program to compose any kind of poetry you wish.

Some examples of computer-generated poems are given below.

"Astronomy"

Taste and touch brighten
spring snow
Weeds brought the refusal
Unbearable

Weeds loitered western stars
The earth catwalks the east
rim
Less vivid

To rest the earth
To have fog-breath
The little western stars fought

"A Lioness"

Upon the shore
All night
Grotesquely
Frigid final a Lioness
says

After a river
Too soon
To feel
Shining meat-eating
emotion
Grinned

"Beyond the Skull"

Happy I cried self-consciously fleeing the city
Green my wife gives to focus and cunning passages
Naked my north room gliding in its homicidal eye
Secretive the streets live gulping with meditations
Inland mountains hurled as trumpets below blades.

Analgesic inland mountains grow to be my head.
Secretive clues watched gulping so that hell
Tarnished my north room to arrange beyond the skull
Cunning passages meditate eyelids of stone.
White western stars had brought wind into city.

The Game of ANIMAL

EXERCISE 1.

For each of the following pairs of animals, write two questions that will distinguish between them. Write one question so that it can be answered "yes" for the first animal in the pair; the other, "no."

DOG	TIGER	MOOSE
HORSE	PUMA	RAM
ELEPHANT	CAMEL	OCELOT
HIPPOPOTAMUS	LLAMA	CHEETAH

EXERCISE 2.

There are many possible ways to distinguish between two things. For each of the following pairs of animals, write seven questions that will distinguish between them.

OSTRICH	PENGUIN
GIRAFFE	GORILLA

EXERCISE 3.

Play ANIMAL on the computer. Teach it the family of animals you developed for this exercise. If members of the class have chosen different families of animals, the computer should be able to identify just about any animal when you are finished. If you want it to remember the animals you have taught it, answer SAVE to the question:

ARE YOU THINKING OF AN ANIMAL?

Also, for a list of all the animals "known" by the computer, just answer LIST to the above question. (See Figure 4-11 on the following page.)

SUMMARY AND CONCLUSIONS

For years, novelists and scientists have been writing and philosophizing about the computer and its effect on man, his privacy, his leisure time, and his outlook. The computer in schools has had a profound effect on teaching mathematics and science; only recently, however, has it begun to be used in the Language Arts curriculum. Nevertheless, its role promises to be significant, principally because of its ability to motivate.

```
ARE YOU THINKING OF AN ANIMAL?


ARE YOU THINKING OF AN ANIMAL?  YES
DOES IT SWIM?  NO
DOES IT HAVE A TRUNK?  NO
DOES IT GO ARF?  NO
DOES IT HAVE RETRACTIBLE CLAWS?  YES
DOES IT LIVE IN THE JUNGLE?  YES
IS IT THE KING OF THE JUNGLE?  NO
IS IT A TIGER?  NO
THE ANIMAL YOU WERE THINKING OF WAS A ?  PANTHER
PLEASE TYPE IN A QUESTION THAT WOULD DISTINGUISH A PANTHER
FROM A TIGER
?DOES IT HAVE STRIPES
FOR A PANTHER THE ANSWER WOULD BE?  NO

ARE YOU THINKING OF AN ANIMAL?  YES
DOES IT SWIM?  YES
DOES IT HAVE FLIPPERS?  NO
IS IT THE LARGEST KNOWN MAMMAL?  NO
IS IT A FISH?  NO
THE ANIMAL YOU WERE THINKING OF WAS A ?  TURTLE
PLEASE TYPE IN A QUESTION THAT WOULD DISTINGUISH A TURTLE
FROM A FISH
?DOES IS HAVE A BONY SHELL WHICH ENCLOSES ITS BODY
FOR A TURTLE THE ANSWER WOULD BE?  YES
```

Figure 4–11. Sample interaction with the computer game, ANIMAL.

After students play some of these games they no longer feel as threatened when they have to write something themselves. Their thoughts and creative talents seem to be freer. The automatic "turn-off" reaction some students have to writing tends to diminish, and all of a sudden creative writing is fun.

There is no doubt that the computer will play a more influential role in the Language Arts class of the future. It may never be able to teach a student the richness of expression, the humor, the drama, or the poignancy to become a great author, but it certainly will help the vast majority of students to express themselves in a more creative, clear, and concise manner.

REFERENCES

Ahl, David H., *Creative Computing in Language and Art.* (Preprint of a forthcoming book, 1974).

Allen, John R., "Current Trends in Computer-Assisted Instruction." *Computers and the Humanities, 7* (1972), 47–55.

Bowles, Edmund A., "Towards a Computer-Curriculum for the Humanities." *Computers and the Humanities, 6* (1971), 37–45.

Harmin, M., Kirschenbaum, H., and Simon, S. B., *Clarifying Values Through Subject Matter: Applications for the Classroom.* Minneapolis: Winston Press, Inc., 1973.

Oakman, Robert L., "Computer Education for the Humanities: Multiple Possibilities at the University of South Carolina." *Proceedings of the Fourth Conference on Computers in the Undergraduate Curricula* (1973), pp. 373–376.

Straub, Detmar, W., Jr., "The Machine and the Rainbow: Computers in the Full Spectrum of the English Curriculum." *Proceedings of the Fourth Conference on Computers in the Undergraduate Curricula* (1973), pp. 216–221.

Thames, Anna Marie, "CAI and English: A Tentative Relationship." *Proceedings of the 1972 Conference on Computers in the Undergraduate Curricula* (1972), pp. 305–310.

COMPUTERS IN THE MATHEMATICS CURRICULUM
by Walter Koetke

INTRODUCTION

Is using the computer in the mathematics curriculum a good idea? Let's first examine what we know. What is accepted as true regarding the use of computers in education?

The computer is here to stay. Today's educators should be, in fact must be, concerned with finding the best ways to use computing facilities. There is no longer any merit in debating whether or not the computer is a useful educational tool. Those who hoped, and those who feared, that computers might take over the task of education were respectively disappointed and pleased. The computer is probably the most valuable single tool available to the education, but it is certainly not the only one and certainly not useful in all situations.

Students who use computers like doing so and are motivated to continue. All of them? Of course not—but far more than any other single technique, device, or idea has been able to intrigue. Student motivation is accepted as one of the keys to successful teaching, and all who have observed students using computing facilities are impressed, if not amazed, by the very evident high level of motivation.

Students with reasonably unrestricted access to computing facilities do astonishingly good work. Because they are unfamiliar with the usual adult constraints of what can and can't be done, their work is often creative and occasionally significant far beyond their classroom. Do you know how many perfect numbers there are? Don't look in a history book—the last three were discovered in 1970 by a high school student with access to computing facilities.

Let's examine some additional facts about the use of computers in education. These might be of even greater interest because they were not always accepted as true. They can all be characterized as former myths that have been demonstrated to be false.

Myth. Only the best students can use the computer. When I had the opportunity to work with first and second graders who were writing, entering, and executing their own programs, this one-time myth was valid, but its validity ends at about that grade level. The myth is particularly noteworthy because exactly the opposite has proven true. Students with a low achievement record benefit more from computer use than those with higher achievement. The reasons for this have not been clearly defined, but motivational differences are probably very important. Most students with high achievement records are already well motivated, while students with low achievement records are not. Since increased motivation is one of the most noticeable effects of computer use, a poorly motivated student will be more affected than one who is already highly motivated.

Myth. Use of the computer in education is a luxury, not a necessity. Dr. Arthur Luehrmann of Dartmouth College is receiving increased support for his contention that "computing is a new and fundamental intellectual resource, in the same sense that reading and writing and mathematics are fundamental intellectual resources." How many educators will categorize reading, writing, and mathematics as educational luxuries rather than necessities?

Let's focus specifically on mathematics for a moment. Can mathematics educators afford to treat the computer as a luxury? Do today's mathematicians and mathematical applications approach the computer as a luxury? Throughout the history of mathematics, the major emphasis of mathematicians has shifted to serve the needs of society best. Today that emphasis is on algorithms and computation. The changing mathematical needs of our society should not be treated as optional luxuries; they should be an important, required part of the curriculum.

One aspect of this changing emphasis can be seen by again looking at the history of mathematics—a history filled by clever men creating ingenious schemes to avoid computation. With the availability of the computer, computation has become very easy rather than tedious and difficult. The ingenious schemes from the past remain ingenious, but their importance is historical rather than contemporary.

Myth. The computer can be readily incorporated into the traditional structure of education. For some types of computer use, primarily problem solving, this myth remains reality. Although problem solving is perhaps the most valuable instructional application, the other possibilities cannot be ignored. The computer makes possible truly individualized instruction, testing, and evaluation—the type of individualization educators have long attempted but never achieved.

The computer provides students with an extension of their own intellect. They can explore, test, analyze and create on a scale never before possible. The work of Dr. Seymour Papert, co-director of the artificial intelligence laboratory at the Massachusetts Institute of Technology, has given some insight into the tremendous educational potential of the computer. Dr. Papert's "mathland" cannot, however, be described as a typical educational environment. The differences are in fact one of the primary reasons for the success of his work. Some specific examples of likely changes in the methods of mathematics instruction are examined in the article "The Impact of Computing on the Teaching of Mathematics," listed in the bibliography.

Where are we now? What changes should be anticipated in the near future? Consider each of these questions as they apply to the computer's different instructional roles.

USES OF THE COMPUTER IN MATHEMATICS INSTRUCTION

Drill and Practice

The question-and-answer nature of drill and practice clearly requires interactive facilities. Batch processing installations will have to omit this application entirely. Programs to provide drill and practice can be implemented on a minicomputer. Although not necessary, a medium-sized computer will often simplify the programming effort required.

The majority of drill-and-practice programs now being used have at least one common property—they are boring. Such criticism should not be applied too harshly, however, for the very nature of drill and practice is boring in all but the cleverest disguise. The more effective drill-and-practice programs seen by the author have been written by students for their peers. When the gimmicks of the student programmer are combined with the educational advice and strategy of a teacher/advisor, effective drill programs can be developed; I have observed them for chemistry, French (verb conjugation), arithmetic operations, special education, and polynomial factoring.

One possible future benefit of computer-controlled drill and practice is that the computer can be programmed to record student progress and report it to the teacher. Although this added dimension has many positive possibilities when the computer is also used in a tutorial role, record-keeping in a strictly drill-and-practice situation has yet to excite many teachers.

Two cautions are in order for the teacher just beginning to use drill-and-practice programs. First, although students can advance through computer-generated exercises at their own pace, the upper limit imposed upon that pace can be rather slow. Terminals and video displays with a data transmission rate of 30 characters per second are far more desirable than the usual Teletype speed for this application. Second, students who need drill and practice usually need it immediately, on a very specific topic for a very limited period of time. Thus to be a generally effective device, the teachers' library requires a substantial programming effort as well as routine maintenance.

Today the use of the computer as drillmaster is limited because effective drill and practice requires that a single student tie up a terminal for long periods of time. As the cost of terminals continues to drop and the rate of data transmission increases, this problem will be less significant. Over the next five years, however, I suggest that most drill and practice will be produced locally by the teacher/student coalitions already described. If you are particularly interested in this application, start programming. I do not believe that truly effective, reasonably priced drill-and-practice programs will be available commercially during the next five years.

Tutorial

Work to date has only begun to reveal the tremendous potential of using the computer to provide the Socratic dialogue type of instruction. If education is ever to be equally available to all children, the computer will be chiefly responsible.

The most effective tutorial instructional network I have seen that is actually implemented and being used is that supported by the Ministry of Education in Ontario, Canada. Their program is designed to meet the remedial and review needs of students studying high school algebra. The province-wide network is used by hundreds of students each day. The performance of all students is monitored, and analysis of the monitored data is part of a continuing effort to improve the program.

A computer tutor clearly requires a medium-sized, interactive facility. Using batch processing would be similar to giving Socrates a half-million-dollar amplifying system so he could be sure his one student heard the questions. Tutorial programs written for minicomputers are primitive and abbreviated at best.

If you are interested in tutorial applications, be patient. As you view existing systems, remember that you are probably seeing a very primitive example which lies only on the surface of a sphere of possibilities. Truly effective use of the computer as tutor must permit the user to respond as he wishes—not with "yes" or "no;" not with the correct answer spelled right or the wrong answer; not with a multiple choice selection; and not even necessarily with an answer; he may have another question. Fantasy, you say? Such systems do exist! But they are not widely available; they require large computers; their development costs are staggering, and their reaction time is often measured in minutes. There are even programs that can understand, analyze, and solve a conventional statement of an algebraic word problem that might be found in high school textbooks. Again, however, you can probably teach your slowest student to solve the problem before the computer finally prints the answer. The computer tutor is the educational hope of the future, but that future remains distant.

Problem Solving

Computer as Problem Solver

If you attend almost any demonstration of computers in education, particularly those of the computer companies themselves, you are likely to believe that this number-crunching application is the main educational use. This is a reasonable conclusion after you are shown an impressive program library that can handle everything from binary addition to double integrals. In reality, however, the application rarely occurs in the mathematics curriculum. Perhaps a radically different mathematics curriculum will eventually make it more prominent, but I foresee no such change in the near future.

Computer in Teaching Problem-Solving

This is where the action is! Allowing students to write, enter, and debug their own programs is unquestionably the primary educational application today—and is likely to remain so for at least the next five years. Teachers can use it to have students review, reinforce, introduce, apply, or create new ideas. Teachers and students are limited primarily by their imagination. The computer is intended to be a tool, and here it is used as intended. The teacher can use the new tool to teach, and the student can use the new tool to learn and explore.

Teaching problem-solving requires no more hardware than a minicomputer. Although interactive facilities are superior, batch processing does not totally exclude this application, but it does severely restrict it. The greatest mistake a teacher can make is to underestimate the scope of the work students will do. The computer is a tool unlike anything else—it really can be an extension of their intellect—and every effort should be made to provide them with an adequate facility.

Testimony regarding the merits of teaching problem-solving have come from advocates of all the instructional roles of the computer. The student who writes a drill program learns more than the student who uses it. The student who writes a tutorial program learns far more than the student who uses it. After one of my students wrote a tutorial program on polynomial factoring, he challenged me to find a factorable polynomial that he would be unable to factor without the computer. I was unable to find one that even offered a significant challenge. Students who write simulations, games, or data analysis programs similarly learn far more than those who use their programs.

The problem-solving application can be used at all grade levels, including elementary. The work of Dr. Papert has been focused primarily on elementary school children, and all are writing their own programs. They not only manipulate numbers, but they also play music, draw pictures, and control robotic turtles through which they can simulate certain aspects of their own behavior.

The largest bulk of curriculum materials available today assumes that the computer is being used in an atmosphere of teaching problem-solving. Unfortunately the available material also assumes that this is being done in an elective course rather than being integrated into the regular mathematics curriculum. Materials for the regular curriculum are coming, and their already overdue arrival should be applauded. In my opinion, elective courses are fun, stimulating, challenging, appropriate, and important; but mathematics teachers can honestly report that their work is serving the needs of society and its children only after the computer and the new mathematical priorities it dictates have been incorporated into the regular mathematics program.

Data Analysis

Including data analysis in the mathematics curriculum is likely to suggest a course in probability and statistics. Such a course—not just a chapter—is long overdue in many secondary schools. There are several reasons for the delay. When presented on a theoretical level, the subject quickly restricts itself to the better students. Realistic problems, even with capable students, are often excessively difficult. When computing facilities are available, both of these obstacles are diminished. Theoretical probability does not become any easier, but having a computer to do much of the computation reduces the amount of theoretical material required. In addition, far more realistic problems can be solved with computer assistance. This application will, however, be difficult to pinpoint as a separate item because in most cases student work is also likely to involve problem-solving as well as the creation and use of simulation programs.

The future should contain a growth in the implementation of data analysis applications. Minicomputers are quite capable of handling most of the programs. Data analysis is one application for which batch processing can be effectively used. Hopefully, the future will also find science and mathematics departments working together on problems of data analysis. Work in the science laboratory is often an ideal application for the techniques of data analysis taught by the mathematics department. Schools with their own computing facilities may even be able to do on-line, real-time data analysis.

Simulations and Games

Computer games fascinate many users but may alienate teachers who do not understand their place. The current state of the art is well documented in David Ahl's *101 BASIC Computer Games*. Certainly games interest students, and most will play a new one two or three times before becoming bored. Where then is their value? The most obvious benefits are obtained by those who write the programs, a task on which many students are willing to spend many hours. When writing programs they are working in the problem-solving role; and the value of this role needs no amplification. Is there value in simply playing a game written by someone else? When play is guided by a skilled teacher, there is indeed value. Analysis of the strategy followed by a program can be both challenging and valuable. Is it the best strategy? Why? How was it determined? Let students play a game in which the computer's strategy is good but not best—then challenge them to determine and defend a better one. Educational benefits of game playing is an appropriate topic for an entire paper.

And then there are simulations. Here again, the one who does the programming learns the most; but simulations can be a valuable teaching aid. They are rarely in the exclusive domain of the mathematics department—they provide a convenient means for involving staff members from other departments in computer usage. The simulations available today have come from two sources: federally funded programs and student-teacher coalitions. The few really good, interesting, and educational simulations today have come from the latter. I expect the future to contain a continually increasing number of good simulations as computing facilities become available to larger numbers of teachers and students.

When considering simulations, teachers should remember two facts. First, a good simulation program is not trivial. The programming and research effort required is significant. Once written, a good simulation is likely to be used only once or twice a year in each class studying that the aspect of the subject, so that the actual use of simulation programs represents only a very small percentage of total computer use. This will not change until a great many more good programs are generally available.

SUMMARY AND CONCLUSIONS

What should this discussion mean to the experienced teacher? to the novice teacher? Hopefully you've reached some conclusions—certainly you're aware that there are more questions than answers when one examines the use of the computer in the mathematics curriculum.

If you choose to leave with only one conclusion, let it be that computer use in a somewhat modified mathematics curriculum is inevitable during the next five years, and that the computer will play an ever greater role in a significantly changed educational structure during the next ten years. The choice for both experienced and beginning teachers is clear: one can help in planning the new mathematics curriculum, or one can sit back and watch things change around him. The precise place for, and best use of, the computer in the curriculum is not know. All teachers are faced with the perhaps unique opportunity of having a significant influence on its formation. There are few experts—the educational use of computers is still a new topic. Learn about computers, then help develop and refine computer applications in that which you've chosen as a career—the teaching of mathematics.

BIBLIOGRAPHY

Ahl, David, *101 BASIC Computer Games*. Maynard, Massachusetts: Digital Equipment Corp., 1973.

———, *Understanding Mathematics and Logic Using BASIC Computer Games*. Maynard, Massachusetts: Digital Equipment Corp., 1973.

EDU, a quarterly magazine of the educational users group of Digital Equipment Corporation, Education Products Groups, Maynard, Massachusetts 01754.

Hewlett-Packard Educational Users Group Newsletter. 10 issues each year. Available through Hewlett Packard, 11000 Wolfe Road, Cupertino, California 95014.

Koetke, Walter, "The Impact of Computing on the Teaching of Mathematics." AFIPS Conference Proceedings, Volume 40, 1972 Spring Joint Computer Conference, AFIPS Press, 210 Summit Avenue, Montvale, New Jersey 07645.

The Mathematical Sciences, A Collection of Essays, edited by the Committee on Support of Research in the Mathematical Sciences. Cambridge: M.I.T. Press, 1969.

Papert, Seymour, *New Educational Technology*, 1973. Available from General Turtle Inc., 545 Technology Square, Cambridge, Massachusetts 02173.

SIGCUE Bulletin of Computer Uses in Education. A publication of the ACM Special Interest Group on Computer Uses in Education, October 1973. ACM, 1133 Avenue of the Americas, New York, New York 10036.

COMPUTERS IN MATHEMATICS INSTRUCTION
by Elizabeth A. Williamson

INTRODUCTION

The era of "new mathematics" has left serious scars on mathematics education. With its emphasis on structure, abstraction, deduction, and precise language, "modern math" has been a major factor in producing a generation of students who have poor computational skills and little or no understanding of mathematics as it relates to the real world, and who have been unable to find a sensible motivation or rationale for the study of mathematics.

Born in reaction to the defects of the traditional mathematics curriculum and a widespread public mathematical illiteracy associated with it, new math was intended to make mathematics understandable. "In the new math," as the social satirist Tom Lehrer quipped, "the idea is to understand what you're doing, *rather* than to get the right answer." Yet the new mathematics movement has failed to alleviate mathematical illiteracy and has not successfully provided motivation for the study of mathematics. Students by and large still have a distaste for the subject.

Furthermore, in glorifying deduction and abstract structure, the mathematicians who created new math curricula attempted to present mathematics as they had come to understand it. But for many students, this approach clashed with a development process of intellectual growth. Piaget* tells us that formal logic in the adult sense of the term (the kind of logic required by the rigorous, deductive approach of the new math) does not even begin to be formed until eleven or twelve years old or, for some children, even later. Thus, for many secondary students, the modern mathematics curriculum was simply incompatible with their stage of intellectual growth. Those who had not yet developed the intellectual structures to deal with the curriculum were in grave difficulty, for most teachers did not consider the personal stage of intellectual growth as a critical variable in the student's ability to succeed with the new math.

Emerging from the new math revolution, mathematics education is once again on the brink of reform. Educators and mathematicians are beginning to realize that in order to head in a proper direction for reform they must first of all be sensitive to the findings of the developmental psychologists. They must create a mathematics curriculum which allows for individual differences in development, and which gives a teacher the flexibility to create learning environments that respect the stage of each learner's intellectual growth. This means that the curriculum has to make options available for learners just beginning to acquire formal logic as well as for those who are more highly developed.

Educators and mathematicians designing a new secondary mathematics curriculum are also acknowledging that they must pay careful attention to motivating the study of mathematics and making it a sensible line of study for the adolescent to pursue. As it is presently taught, mathematics does not and perhaps should not appeal to ninety-eight percent of the students. Morris Kline points out that it is now presented as an esoteric study of abstractions, entirely intellectual in its appeal. It may have some emotional values for the creative mathematician, but certainly not for the student.

At least two related strategies can be suggested for motivating a study of mathematics. One is to relate it to the real world, and indeed even to demonstrate or let the student discover how the mathematics arises from and describes real world experiences. Motivation for the nonmathematician cannot be mathematical.

*Jean Piaget, *To Understand is to Invent* (New York: Grossman Publishers, 1973), p. 50.

> It is pointless to motivate complex numbers (for example) for the general student by asking for solutions to $x^2 + 1 = 0$. Since nonmathematicians don't care to solve $x - 2 = 0$, why should they care to solve the former equation? Calculus texts "motivate" many of the concepts and theorems by applying them to the calculations of areas, volumes, and arc-lengths. But these are also mathematical topics and the fact that the calculus enables us to calculate them does not make the subject more engrossing to the nonmathematicians.*

One natural motivation for mathematics is the study of real and largely physical problems. Historically speaking, nearly all the major branches of mathematics arose in response to such problems. For the lay person, mathematics should be a means to an end, and one should be able to use the concepts and reasoning to derive results about real things.

> The use of real and especially physical problems serves not only to motivate mathematics but to give meaning to it . . . the ellipse is not just a peculiar locus but the path of a planet or a comet. Functions are not sets of ordered pairs but relationships between real variables such as the height and time of flight of a ball thrown up into the air, the distance of a planet from the sun at various times of the year, and the population of a country over some period of years. Functions are laws of the universe and society. Mathematical concepts arose from such physical situations or phenomena and their meanings were physical for those who created mathematics in the first place.†

Much lip service was given to relating mathematics to the real world during the era of the new math. But "applications" were for the most part limited to the word problem section at chapter end, in which the student was asked to use some abstractions he has "learned" to solve a contrived and artificial "problem" about real things. Rarely were students given the opportunity, the time, and the support to make substantial sense of and truly understand the mathematics in terms of the real world.

For the most part, the majority of mathematics students acquired boring, meaningless (to them), and easily forgotten skills. Many succeeded simply because they became proficient at symbol manipulation, but they in fact lacked any understanding of what they were doing. The psychologists would tell us that this is the result of moving too quickly to a highly abstract level without taking care to ground the ideas in real world experiences, or at least in experiences at a much lower level of abstraction. Thus a major strategy to be used in math education is to capitalize on the intimate relationship between mathematics and the world of real experience.

A second strategy to solve the problem of motivation is implied by the first: it is to use a "constructive" approach to mathematics. With this approach, students themselves do the discovering of relationships and the creating and building of theorems and proofs. In mathematics as it is currently taught, the student is asked to accept from outside an already organized intellectual discipline which he may or may not understand. The constructive approach would put the learner in a context where autonomous activity would lead him to discover relationships and ideas by himself. This is not to say that the student must rediscover all of mathematics, but only that when feasible, students ought to be given the chance to arrive at some of the major mathematical ideas on their own. Traditional and modern math textbooks both give students the impression that mathematics just somehow came to be. That mathematics is a very human endeavor escapes them. And if they do realize that the theorems and proofs in a geometry book, for examples, were written by people, students are still deluded into thinking that the mathematician who wrote them reasoned directly and unfailingly to his conclusions. Mathematicians know that most creative work in mathematics is riddled with false starts, guessing, and dead ends. But students do not know this, and thus are reluctant to become involved in autonomous activity related to mathematics. They see no ethic of trial and error or of exploration on the pages of their math tests. Thus, the constructive approach both introduces the student to new concepts and encourages him to become involved in the process of building mathematical theories of his own.

USES OF THE COMPUTER IN MATHEMATICS INSTRUCTION

Any role that is defined for the computer in mathematics education necessarily emerges from the basic

*Morris Kline, *Why Johnny Can't Add* (New York: St. Martin's Press, 1973), p. 105.

†Ibid., p. 153.

philosophy and orientation of the mathematics curriculum itself. Computer use over the past several years has been shaped and controlled by the philosophy and content of the new math curriculum. Thus, students have been asked to program algorithms in order to learn them, because learning algorithms was valued as an end in itself in the curriculum. Since the curriculum itself was formalistic, the computer was used to support that formalism. Drill-and-practice and tutorial routines have been popular, again because they supposedly helped students master the facts and the algorithms. Students have not often pursued independent investigations using the computer, for independent investigation was not, for the most part, valued or encouraged in the secondary mathematics curriculum. And only the most innovative teachers used secondary mathematics as a context for developing mathematical models. This approach did not fit the general curricular orientation, and teachers who used it had to be willing to go off the beaten curricular path.

If the new directions for reform outlined at the beginning of this article influence the mathematics curriculum of the future, our view of appropriate roles for the computer must also be influenced.

Drill-and-Practice and Tutorial

The role played by the computer as drillmaster or tutor as it is currently conceived and implemented would have little use in the new curriculum because it makes no contribution to developing an ethic of mathematics instruction that provides motivation for learning mathematics. Nor does it foster real understanding. Rather, by implication it blesses and encourages memorization without understanding and the development of skill in symbol manipulation. It divorces theory from applications, and isolates drill and practice in using algorithms from any context in which the drill and practice contributes to some other goal. If drill and practice is needed, it can probably be done as effectively without the computer as with it. But ideally, we should attempt to raise it to a problem-solving level,* a strategy which will be explained in the following discussion.

*The notion of drill and practice at a problem-solving level was [illegible] Robert Wirtz in his work on curriculum [illegible] elementary school children. It is found [illegible] applicability, however. See Robert Wirtz, [illegible] Practice at the Problem Solving Level—An Alternative (Washington, D.C.: Curriculum Development Associates, 1973).

Problem Solving

Use of the computer as a problem-solving tool will take on new importance under the new directions for the curriculum. One of the major concerns in the earlier discussion of curricular reform was that mathematics be more authentically related to the real world, both in terms of creating learning situations in which mathematical concepts could arise from direct experience and in terms of applying newly discovered mathematical ideas to real world problems. In addition to the highly formalistic approach of the modern math curriculum, there were probably two other major problems which heretofore prevented integration of mathematics and real experience. One was lack of full teacher understanding of the relationship. A second was the lack of appropriate tools with which to make real world investigations, because once we become enmeshed in real problems of significance, calculations often get complicated, messy, and are prone to error.

The computer provides a powerful tool which can truly enable building the bridges between mathematics and the real world. A study leading toward the concept of function, for example, might begin with observations of everyday phenomena which exhibit functional relationships (e.g. the weight of a package and the cost to mail it; caloric intake and body weight; the circumferences of circular objects and their diameters). When some understanding of the notion has been developed, we may want to look at functions that describe population growth. After appropriate background work, we might look at the function derived by Pearl and Reed for the growth of population†:

$$y = \frac{L}{1 + a(2.718)^{kt}}$$

where y is the population of a country t years after a fixed data, a and k are numbers whose values depend on the region to which the formula is applied, and L is an upper limit imposed by physical conditions on the population of a region or country. Students can use the computer to generate points for plotting a graph of this function, and for exploring the variations caused by changing parameters. They can discuss the meanings in real terms of the graph and the

†See Morris Kline, *Mathematics and Western Culture* (New York: Oxford University Press, 1964).

variations. Given values for L, t, a, and k which make the formula specific to the United States:

$$y = \frac{197.27}{1 + 67.32(2.718)^{-.0313t}}$$

students might again use the computer to generate data for plotting a graph which depicts the growth of the population of the United States over a period of years. Populations predicted by the formula for specific years can be compared with census reports for those years—leading to a discussion of the nature and limitations of a mathematical model. Population growth for other countries can be studied. Other functions derived for population growth might be found, and students could use the computer to compare the predictions of each. Many other questions and directions of study could be pursued. Throughout this entire process, the mathematics is being tied to something that is very real in the student's experience -the number of people in his world. Significant investigations are taking place, and it is the computer which enables these to occur. Without the computer, it would not be feasible for a student to conduct this sort of investigation. Done by hand, the calculations would be unbearable.

In the case just described, the program could be written by a student or group of students, or it could be supplied as a "canned" program. Here a significant point can be made. If students write the program themselves, they are in a context where writing the program is not an end in itself, nor is it done only to learn an algorithm or reinforce a fact. The end of the activity is to get the student to use a mathematical function to investigate population growth. Yet in the process of achieving that end, we can also work toward accomplishing other objectives, some of which might be skill in programming, practice with exponentials and graphing, or building understanding of the notion of function. But here, these objectives are achieved as by-products of engagement in the principal activity of investigating the nature of population growth. Learning the techniques of programming or graphing exponentials are no longer ends in themselves; they have been raised to a problem-solving level and become means to an end that makes sense in the students' world of real experience.

Many investigations of this type can be designed and integrated with computer use. Especially helpful in planning endeavors of this sort are such books as Morris Kline's *Mathematics in Western Culture* and James Newman's *The World of Mathematics*.

The new curricular and pedagogical considerations help us to further evolve our view of the computer as a problem-solving tool. In particular, the computer can play a significant supportive role if we begin to take a constructive approach that allows and encourages students themselves to discover ideas and build theorems and proofs. The approach can be used to get at specific mathematical content and also to involve students in the process of making independent investigations by observing data, creating hypotheses, and building theorems.

For example, one pervasive concept in secondary mathematics is the relationship between an algebraic representation of a set of points (an equation) and th[illegible] representation of the same set of poin[illegible] the equation). Understanding the rela[illegible] an equation and its graph is a funda[illegible]ical skill, which assumes even greate[illegible] when we put more emphasis on [illegible] of mathematics. For simple linear and [illegible] equations, this relationship is easy to demonstrate and illustrate. But once the equations become more interesting, tedious calculations are necessary to generate enough points to get an accurate picture of the graph of the equation. The calculations involved in determining points which satisfy equations describing the conic sections, for example, get very messy, are exceptionally prone to err[illegible]nd usually involve the use of a square root table[illegible]s a consequence, students usually do not themselves "construct" graphs from the points generated by the equations and make their own conjectures about the nature of the graphs which result from certain types of equations. They are deprived of extensive experience with graphing a variety of equations and the opportunity to make pictorial associations with the various types of equations. In short, there is not much chance for them to make sense of the relationships. In many cases, students resort to learning the symbol manipulation "tricks" which get the equations into specific "forms" and then learn the "rules" for identifying the type of graph that will result from specific equations. But in fact, they lack any real understanding, for they are forced to use pure abstractions too quickly. For many students, this jump does violence to their intellectual development, and the psychologists would tell us that moving so fast to a high level of abstraction is a very good way to stifle concept development for all learners.

On the other hand, if a student can write or have access to a computer program which will generate sets of points that satisfy an equation describing a conic section, he can use the computer-generated data to plot as many graphs as he wants or needs to. He can compare the equations with the graphs, and he can begin to make some hypotheses about relationships between the equation and the shape and position of the graph. He can test his theories and refine them. In this way, the student is able to make a more extensive and understandable study of the relationships between an equation and its graph than was ever feasible without the computer, and he can build an experience base which will enable him to understand the transition to an algebraic analysis of the equations. But most importantly, the student will himself have discovered the relationships between the equations and their graphs, and will have become involved in the process of creating mathematics.

Another major role for the computer emerges when we think about using a constructive approach to mathematics education. We can begin to think of the computer as a problem-suggesting tool as well as a problem-solving tool.

One way to involve students in creating mathematics by building theories is to put them in learning situations where, while there may be no specific content we hope they will discover, they can encounter data which suggest hypotheses that they can formulate, test, and revise, and from which they can build theorems. The computer can be of enormous value in creating such problem-suggesting situations. An illustration will clarify the idea.

Many secondary students are familiar and comfortable with real numbers in the raw, even though they are still not able to deal easily in a formal logical manner with algebraic statements involving real variables. Thus, they can observe data or tables of real numbers and make hypotheses about patterns they see, even though the same hypotheses handed down in the abstract would probably baffle them.

Project REACT developed a computer program which enables such independent investigations. This program (1) lists, tallies, and sums the factors for any given positive integer and (2) lists, tallies, and sums the factors of every integer between any two given integers a and b inclusive. Even the tables of numbers generated by listing, tallying, and summing the factors of numbers from 1 to 30 is replete with patterns and relationships which students can discover. The data themselves suggest problems to explore and questions to pursue. It is in this sense that we can view the computer as a problem-suggesting tool: it gives the student access to data that would never be available to him if he had to generate them himself.

Use of the computer as a problem-suggesting tool also promotes individualization in one of the best senses of the term. Several different learners can use the same data base, and each can pull from it problems and questions at his own level of sophistication. Some learners will make only the most obvious observations and hypotheses, and develop only intuitive proofs, while others will develop intricate theories and be able to present formal, deductive proofs. But all will be engaged in the important process of creating mathematics, and will have the freedom to move in a direction which respects their level of intellectual growth. Many computer programs of this type can be designed by both teachers and students, and some are even available as canned programs.*

Data Analysis

Use of the computer for statistical data analysis is certainly appropriate in mathematics education, especially as we look toward to the future with its emphasis squarely on applications. A program such as the Statistical Analysis Package (SAP) described in the *Uses of the Computer In Instruction* booklet would certainly merit use in the mathematics curriculum. Applying statistics to the real world is one of the most familiar applications of mathematics, and also one that can generate a high level of interest among students.

*See in particular the REACT Mathematics Units (Portland, Oregon: Northwest Regional Educational Laboratory, 1972). Another good example of a program which supports individualized investigations is one suggested by John Williamson in his article "A General Structure for the Study of Prime Numbers," *The Mathematics Teacher, 60* (March 1967). Williamson creates some new definitions and points a direction for investigating the distribution of prime numbers in a variety of domains. A rather simple computer program is used for the investigation. The data generated are rich with patterns and suggest many theories and questions which could be formulated by even average secondary students.

Simulation

Computer-based simulations and games also have real relevance to the mathematics curriculum. Once again, the computer plays the role of an enabler, for by developing mathematical models as the basis for computer simulations, students can get into the process of experimenting with their models. The computer allows the otherwise static model to come to life and allows the student to explore the implications of his model and refine it appropriately.

One should not assume that sophistication in higher mathematics is prerequisite to building models. Even young learners can build models which, though elementary and crude, capture some essential features and relationships of a real situation. A wide variety of suggestions at various levels of difficulty for models which could be built by secondary students can be found in The Engineering Concepts Curriculum Project publication *The Man-Made World*. The development of mathematical models and eventually of simulations could be for some students a natural outgrowth of a mathematics curriculum which focused sharply on the relationship of mathematics to the real world and used a constructive approach.

Some Concerns

Several major concerns must be voiced at this point in the discussion. The first relates to the nature of the materials that will be offered to support computer usage. In my opinion, if computers are to be fully utilized, the possibilities must arise from the context of a total curriculum rather than from isolated appendages to whatever else happens to be in use by the teacher at the time. A good curriculum must have a clear conceptual design, appropriate teaching methodologies, and an underlying theory of learning with which the concepts, content, and teaching methodologies are consistent. It should also be "unfinished" in a very sophisticated sense.* An unfinished curriculum has a conceptual design and a content backed by scholarship in the academic discipline which are compatible with a creditable theory of learning. It is not bounded by a rigid scope and sequence; rather, its conceptual design is clear enough to enable teachers and scholars to adapt, modify, and go beyond the original design. It gives teachers a valid and usable framework within which they can develop, test, and use materials and activities. An unfinished curriculum provides teachers with a model curriculum and invites and encourages them to continue its development in accordance with their own talents and teaching situations. Such a curriculum enables teachers to look at their own developments in a broader context and gives them a basis for judging the effectiveness of these developments. If such a curriculum could be developed for secondary mathematics, it would allow for continuing developments in computer usage, but it would help teachers know how to use these developments and still preserve the integrity of the curriculum.

Directly related to this is a concern for teacher education. Use of the computer as conceived in this paper requires a revised view of teaching and learning, and in many cases requires of teachers new intellectual skills in mathematics. There can be no successful implementation of a curriculum which expects these changes on the part of teachers without a strong commitment to teacher education. Ideally, the teacher education should be curriculum-based, and should be an integral part of the dissemination effort.

*The notion of an unfinished curriculum was first conceived of by Francis R. Link, Senior Associate at Curriculum Development Associates, in characterizing the curriculum as *Man: A Course of Study*.

SUMMARY

The roles defined for the computer in mathematics education are necessarily determined by the philosophy and orientation which underlie the curriculum itself. New directions in mathematics education now emerging call for a new look at roles for the computer. Increased emphasis on the tie between mathematics and the real world suggests a significant role in allowing for investigations of significant real world problems. A trend toward taking a constructive approach to mathematics also opens new roles for the computer, both as a problem-solving tool and a problem-suggesting tool. Use of the computer in data analysis and simulations and games also takes on new importance for a curriculum which highlights the relationship of mathematics to the real world. If computer usage is integrated with a total curriculum and bound to a strong teacher education commitment, it can become an active agent of continued curricular renewal.

REFERENCES AND SOURCES

The following are sources for further information about the theory and practical content of this article and for suggestions for the kinds of computer programs that have been highlighted here.

Dorn, William, *Mathematics and Computing.*

Engineering Concepts Curriculum Projects, *The Man-Made World* (New York: McGraw-Hill Book Company, 1971).

Kline, Morris, *Mathematics in Western Culture.* (New York: Oxford University Press, 1964).

———, *Why Johnny Can't Add: The Failure of the New Math.* (New York: St. Martin's Press, 1973).

Link, Frances R., "The Unfinished Curriculum." *Compton's Yearbook*, 1973.

Piaget, Jean, *To Understand is to Invent.* (New York: Grossman Publishers, 1973).

Newman, James, *The World of Mathematics.* (New York: Simon and Schuster, 1956).

REACT Mathematics Units. (Portland, Oregon: Northwest Regional Educational Laboratory, 1972).

Williamson, John N., "A General Structure for the Study of Prime Numbers." *The Mathematics Teacher, 40* (March 1967).

Wirtz, Robert W., *Drill and Practice at the Problem Solving Level–An Alternative.* (Washington, D.C.: Curriculum Development Associates, 1973).

COMPUTERS IN MUSIC EDUCATION by Linda Borry

INTRODUCTION

Many people are beginning to acknowledge the value of the computer in math, science, and business education classes. Some see possibilities for it in social sciences, industrial arts, and home economics. Very few, however, have ever considered its place in the humanities; indeed, to some the concept of computers in the humanities is a contradiction—they see the computer removing the human element from the humanities. In this article I hope to shed some light on what is being done and what could be done with the computer in the area of secondary school music education.

First, we should review the prevalent secondary school music course offerings and their apparent goals. The offerings can be divided into three categories: performance groups, general music, and music appreciation and theory. The performance groups—band, orchestra, choir, and small subsets of these—seek to give the student skill in a performing medium; to develop his understanding of various musical elements, structures, styles, and periods through performance of works by selected composers; and to give him the deep appreciation of musical performances which can be gained only by having participated in a performing group. The goals of the general music class are similar, but there is less emphasis on skill in performance and more on listening skills and studying *about* music. Theory classes, like general music classes, do not place the emphasis on performance skills but delve deeply into the elements and structure of music.

USES OF THE COMPUTER IN MUSIC INSTRUCTION

How does, (or can.) the computer fit into the three-part music curriculum? I see it as having impact in all five of its instructional roles: drill, tutorial, problem solving, data analysis, and simulation.

Drill and Practice

Music, perhaps more than other disciplines, requires skills that must be learned by drill and practice. All good musicians remember spending seemingly endless hours practicing scales and other rudiments essential to good performance. Although little has been done to utilize the computer to monitor practice sessions, it has been used easily and successfully for drilling ear-training skills.

How is this done? Let me give an example. If a teacher is interested in drilling students in interval recognition, he could prepare a set of cassette tapes to be used in conjunction with a computer program.* Each tape might contain ten exercises of five notes each. In the easiest exercises the notes are separated by intervals of a major second; slightly more difficult exercises have either major seconds or thirds; and ultimately the most difficult offer a wide variety of intervals. After creating the tapes, the teacher inputs the notes into the computer, along with the name of each student who is to use the program.

What does the student do and see? After logging in to the computer he receives information telling which tape should be used that day. Then, from one to four notes of the appropriate exercise are printed, with the missing notes indicated by blanks. After listening to the tape, the student fills in the missing notes, getting three chances to correctly identify each note, after which the computer responds with the correct answer. In addition to presenting the drill and giving instant feedback, the computer keeps records on student progress, noting the frequency of errors for the various intervals. Based on these records, it can prescribe appropriate supplementary or remedial tapes for individual students.

*I here describe a hypothetical and fairly sophisticated computer program for drilling students on interval recognition.

This type of program could easily be modified to drill such things as rhythmic or harmonic pattern recognition.

Some advantages of computer drill over more conventional drills are that it provides more opportunity for individualized curriculum to meet students' needs, allows more class time to be devoted to other activities, and provides accurate and up-to-date records of student progress and difficulties which can facilitate individualized student-teacher interaction and student grouping.

Tutorial

As in many other fields, the computer can be used advantageously in music education in a tutorial mode—that is, to teach new skills or concepts—for example, form recognition, elements of interpretation (phrasing, dynamics, etc.), and such aspects of theory as harmonic analysis.

You may wonder how an average music teacher with no computer background can computerize material to take advantage of the capabilities offered. The answer usually is to use an author language which is available on the system. Author languages vary greatly in both flexibility and ease of usage, but for the most part they are fairly simple and straightforward and enable the person with little or no computer background to create good tutorial materials.

Problem Solving

A third mode of computer usage in music education is problem solving—using the computer both to do data manipulations which by hand are cumbersome and tedious, and to teach problem-solving skills through programming.

Anyone who has studied music theory remembers the headaches involved in working with the components of 12-tone music; the row, its inverse, retrograde, inverse of the retrograde, and the various transpositions. Deriving these is a natural task for a computer. Given as input any tone row, the computer can respond with all the related elements in the time it takes to do the printing.

Teaching problem-solving skills by having students write computer programs has interesting applications in music, especially theory. As an example, a fascinating interdisciplinary study can be made by exploring the relationship between mod 12 arithmetic and both 12-tone music and transposition. To do

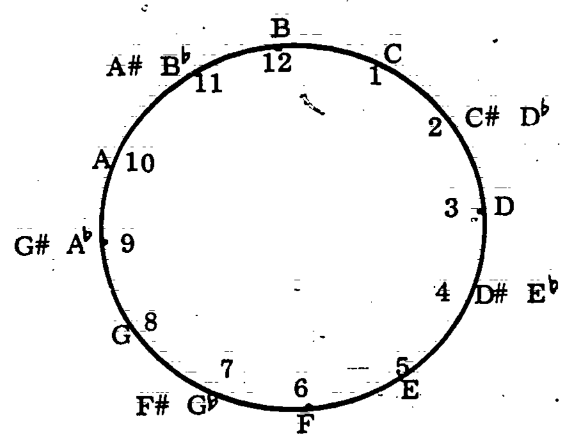


Figure 4–12. Mathematical model of the 12 chromatic pitches.

this it is necessary to create a mathematical model of the 12 chromatic pitches, as in Figure 4–12. The pitches are arranged in order around the face of a clock, with each pitch corresponding to a number. This model can be used in transposition, as follows. Suppose a melody is to be transposed up a major second. To determine the transposed notes, simply locate the original note on the wheel and move two positions in a clockwise direction: an F becomes a G, an B^b becomes a C, and so on. With a very rudimentary knowledge of BASIC or some other computer language, students can write a simple program to do transposition, thus reinforcing their understanding of the procedure and at the same time learning concepts of mod 12 arithmetic.

```
10   PRINT "NUMBER OF HALF STEPS UP";
20   INPUT B
30   READ A
40   IF A=0 THEN 110
50   A=A+B
60   IF A <= 12 THEN 80
70   A=A-12
80   PRINT A;
90   GOTO 30
100  DATA 1,1,8,8,10,10,8,6,6,5,5,3,3,1,0
110  END
```

Figure 4–13. Listing of a sample transposition program.

Figure 4-13 shows a transposition program listing where the numbers in the DATA statement correspond to the notes of "Twinkle, Twinkle, Little Star":

Figure 4-14. Musical Notation of DATA Statements in Figure 4-13.

After asking the user where to transpose the piece and receiving a response, the computer takes the "notes" from the DATA statement and one by one adds the appropriate interval to each, checks to see if the new note is greater than 12 and if so corrects it, and finally prints out the transposed note. In the sample runs (Figure 4-15), all typing done by the user is underlined.

```
RUN

NUMBER OF HALF STEPS UP? 2 — Key of D
 3       3       10      10      12      12      10      8       8       7       7       5
 5       3
DONE
RUN

NUMBER OF HALF STEPS UP? 10 — Key of B♭
 11      11      6       6       8       8       6       4       4       3       3       1
 1       11
DONE
```

First run changed to notes

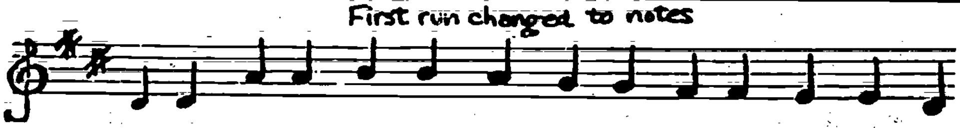


Figure 4-15. Sample run of program listed in Figure 4-13 and the musical notation of the first run.

Another interesting application of this model is in determining the inverse of a 12-tone row. The inverse is normally formed by inverting all intervals in the row: where the row goes up a 4th, the inverse goes down a 4th, and so on. When inverting a row by means of a tone wheel, one simply counts the number of positions the row moves in a clockwise direction and then finds the next note of the inverse by moving that many positions counter-clockwise. It is a relatively simple programming problem, and by writing the program, the student not only learns about the inverse of a tone row, but also begins to unravel some of the mysterious intertwining of math and music.

Data Analysis

A fourth mode of computer usage is data analysis, and here too applications may be found in music. It is possible to put an entire musical score into notation that can be handled by a computer and analyze it for any number of things: melodic, rhythmic, and harmonic patterns; phrase length and construction;

66.7% of all D's were followed by D's.
33.3% of the D's were followed by A's.

	D	D#	E	F	F#	G	G#	A	A#	B	C	C#	
D	66.7	0.0	0.0	0.0	0.0	0.0	0.0	33.3	0.0	0.0	0.0	0.0	3
D#	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
E	50.0	0.0	50.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2
F	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
F#	0.0	0.0	50.0	0.0	50.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2
G	0.0	0.0	0.0	0.0	50.0	50.0	0.0	0.0	0.0	0.0	0.0	0.0	2
G#	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
A	0.0	0.0	0.0	0.0	0.0	33.3	0.0	33.3	0.0	33.3	0.0	0.0	3
A#	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
B	0.0	0.0	0.0	0.0	0.0	0.0	0.0	50.0	0.0	50.0	0.0	0.0	2
C	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
C#	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0

Rows indicate the number of times each note occurred.

Figure 4-16. Sample run of a music data analysis program.

use of dynamics; and others. What is more, if numerous compositions are stored in computer notation, astounding comparisons can be made. The technique is valuable for trying to ascertain authorship of a composition: the computer compares data on a questioned composition to data on pieces known to have been written by a certain composer.

An example of using the computer for musical analysis could be a problem in which you are trying to determine melodic characteristics of a certain composer's music. One method might be to set up a melodic probability chart indicating the frequency with which any note is followed by another note. The computer can take as input the melodic line of several compositions written by a composer, analyze them, and produce as output a chart like the one in Figure 4-16. Generating charts on a number of composers and comparing them by computer can produce interesting results.

Simulation

The fifth and, in my opinion, most exciting use of computers in music, is simulation, in which the computer is used to write music. An interesting project offering valuable learning experiences is to have students "teach" the computer to write simple two-part compositions which, although not particularly interesting, are fairly pleasing to the ear. How might a student go about doing this?

First he would learn about the random number generator.* Then he would construct some type of musical model, perhaps by assigning the numbers 1-25 to the notes from *g* to *g2* and—generating random numbers in the range of 1-25—have the computer create a melody. Most such melodies are simply too random to be pleasing to the ear. But after experimenting with several, students may decide to place controls on the note selection process. One control could state that only notes in the G major scale are acceptable; a further step might involve limiting the number, size, and direction of the skips in the melody; and so on. Some students might even come up with the idea of using melodic probability charts like those described earlier. Some students may stop after producing a few acceptable melodies, but others will experiment with adding rhythm and harmony, all the time learning more about music and composition. In the entire process, the computer plays the role of a stupid but obedient servant who has no personal taste in music.

The sample in Figure 4-17 was written by the computer after much learning had occurred.

A student who works on a project of this nature not only has fun and learns theory; he also develops an appreciation for the human mind and its creative talents and for the works of good composers, who are able to break many of the "rules" and yet produce inspiring music.

*Use of a random number generator is comparable to rolling a die or spinning a spinner.

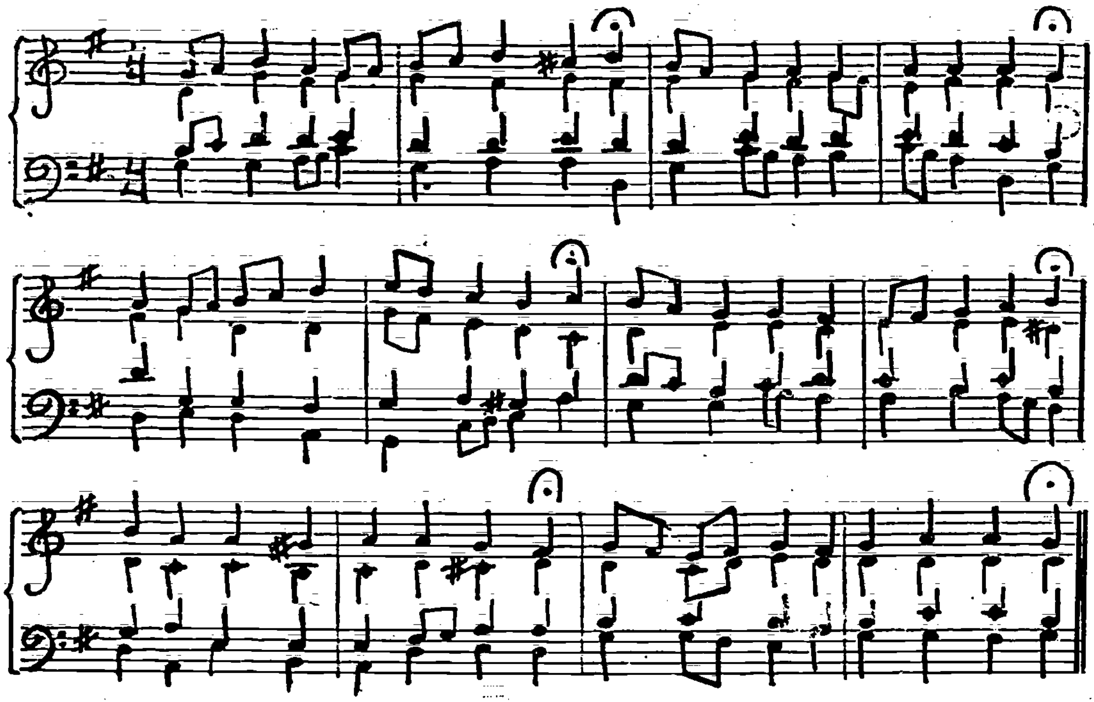



Figure 4-17. Notation of a composition written using a computer simulation program.

SUMMARY AND CONCLUSIONS

The computer, with its capabilities in drill, tutorial, problem solving, data analysis, and simulation can be an asset to music education of all types, but is of particular value in music theory. Although few music teachers have investigated its potential, those who have generally express positive attitudes. I foresee the bulk of computer usage in music education lying in the areas of drill and tutorial; some, however, will avail themselves of the opportunity to use the computer in the other capacities, especially as computer literacy among students increases. The possibilities are exciting!

REFERENCES

For further reading:

Peoples' Computer Company, P.O. Box 310, Menlo Park, CA 94025, (1972-77 issues)

Lomax, J. D., *Computers in the Creative Arts.* National Computing Centre Ltd., Quay House, Quay Street, Manchester M33HV, England.

COMPUTERS IN PHYSICAL EDUCATION by Jim Sydow

INTRODUCTION

The computer is described as a machine which assists us in the menial mental tasks, thus expanding our abilities and capabilities. The computer is fast, accurate, and stupid while man is slow, inaccurate, and brilliant. Although its application to the world of physical activity is not readily apparent, the computer can serve as a viable instructional tool in the process of physical education. The purpose of this article is to explore various computer applications which either could be or are being accomplished in the physical education curriculum.

Typically, education in a physical activity incorporates not only the actual play, but instruction on rules, optimal strategies, general feel of play, physical as well as other benefits, and skill development. While strategy execution and skill development are best left to direct experience in the activity, the area of developing an understanding of the rules, strategies, and benefits of these activities can make significant use of computer-based instructional applications similar to those used in other areas.

USES OF THE COMPUTER IN PHYSICAL EDUCATION

Drill and Tutorial

Not all modes of computer usage are applicable in the physical education curriculum. Drill and tutorial uses appear to offer few benefits. Rules for an activity may be learned by computer drill, but the amount of time required to accomplish this for a class is probably too great to make it a viable use.

Individualized computer drill on exercises and their physiological benefits can be useful for learning activities which strengthen all areas of the body for maintaining life-long physical fitness. For this instruction to be effective, however, an additional step is required—a transition must be made from the mental knowledge to the physical performance of the activity.

Problem-Solving

Computer problem-solving uses fall into two categories. First are the prewritten programs designed to solve a particular problem, such as scheduling games; these may be used continually as the same situation is confronted. Figure 4-18 shows a sample run of a program which generates round-robin schedules for an entered number of teams.

```
GET-$SCHDLI
RUN
SCHDLI

ROUND ROBIN SCHEDULE                      PERIOD 1
INPUT # TEAMS?6                           GAME      HOME AWAY
0    1    2    3    4    5                  1         1    2
                                            2         5    3
1    0    3    4    5    2                  3         6    4

2    3    0    5    1    4                PERIOD 2
                                          GAME      HOME AWAY
3    4    5    0    2    1                  4         3    1
                                            5         6    2
4    5    1    2    0    3                  6         4    5

5    2    4    1    3    0                PERIOD 3
                                          GAME      HOME AWAY
                                            7         1    4
                                            8         2    3
                                            9         5    6

                                          PERIOD 4
                                          GAME      HOME AWAY
                                           10         5    1
                                           11         4    2
                                           12         3    6

                                          PERIOD 5
                                          GAME      HOME AWAY
                                           13         1    6
                                           14         2    5
                                           15         3    4

                                          DONE
```

Figure 4-18. Sample run of problem-solving program used in physical education.

Other types of management assistance can be offered the physical education teacher—for example, using the computer to schedule school facilities with the physical education classes. Although these are not directly instructional uses, they support and assist the teacher.

The second way computers can be used in problem solving is to involve the student in writing problem-solving programs for sports and other recreational activities. Many benefits are derived when the student, through programming, must teach the computer exactly what to do. The program listing and run beginning below is a student-written bowling scoring program.

Data Analysis

A key interest factor in many competitive game-type activities is the myraid of statistics maintained. The process of storing, accumulating, and calculating data and printing formatted reports is much more comfortably handled by the computer than by the busy teacher. The information received from the data may be used not only for dissemination but for analysis of a student's performance in a particular activity. The program on individual basketball statistics is part

```
10  DIM A [25]
20  FOR X=1 10 23 STEP 2
30  LET A[X] =INT (15*RND(0))
40  IF A[X]< 4 or A(X)> 10 THEN 30
50  IF A[X]=10 THEN 110
60  LET A[X+1]=INT(15*RND(0))
70  IF A[X+1]>9 THEN 60
80  IF A[X] + A[X+1]> 10 THEN 60
90  NEXT X
100 GOTO 130
110 LET A[X+1]=30
120 GO TO 90
130 PRINT "FRAME" "1ST BALL","SCORE"
140 LET S=0
150 LET X=F=1
160 PRINT F,
170 IF A[X]=10 THEN 270
180 PRINT A[X],
190 IF A[X]+A[X+1]=10 THEN 330
200 PRINT A[X+1],
210 LET S=S+A[X]+A[X+1]
220 PRINT S
230 IF F=10 THEN 360
240 LET X=X+2
250 LET F=F+1
260 GOTO 160
270 PRINT "STRIKE","***",
280 IF A[X+2]=10 THEN 310
290 LET S=S+A[X]+A[X+2]+A[X=3]
300 GOTO 220
310 LET S=S+A[X]+A[X+2]+A[X+4]
320 GOTO 220
330 PRINT "SPARE",
340 LET S=S+A[X]+A[X+1]+A[X+2]
```

(continued)

Figure 4–19. Listing and run of student-written bowling scoring program.

```
350 GOTO 220
360 IF A[19]=10 THEN 390
370 IF A[19]+A[20]=10 THEN 460
380 GOTO 540
390 IF A[21]=10 THEN 430
400 IF A[21]+[22]=10 THEN 510
410 PRINT "EXTRA BALLS:" A[21],A[22]
420 GOTO 540
430 IF A[23]=10 THEN 530
440 PRINT "EXTRA BALLS:  STRIKE:,A[23]
450 GOTO 540
460 IF A[21]=10  THEN 490
470 PRINT "EXTRA BALL:"A[21]
480 GOTO 540
490 PRINT "EXTRA BALL:  STRIKE"
500 GOTO 540
510 PRINT "EXTRA BALLS:"A[21],"SPARE"
520 GOTO 540
530 PRINT "EXTRA BALLS:  STRIKE    STRIKE"
540 PRINT
550 PRINT "END OF GAME"
560 END

RUN

FRAME          1ST BALL       2ND BALL       SCORE
  1            STRIKE         ***             24
  2            STRIKE         ***             43
  3              4            5               52
  4              8            SPARE           71
  5              9            0               80
  6              8            SPARE           98
  7              8            SPARE          116
  8              8            0              124
  9              5            0              129
 10              8            1              138

END OF GAME

DONE
```

Figure 4–19. Continued.

of a package of athletic statistics programs for a variety of activities. (See figure 4–20.)

The 16 data items for the respective players are entered in the following order:

3 Defensive categories
3 Neutral categories
3 Minus categories
5 Offensive categories
(Assists, FGM, FGA, FTM, FTA)
2 Time categories
(Quarters, Games)

The three specific items to be measured in the first three general categories are left to the discretion of the user.

Calculated Statistics are derived in the following manner:

D = Total Defense = D1 + D2 + D3
N = Total Neutral = N1 + N2 + N3
M = Total Minus = M1 + M2 + M3
O = Total Offense = Assists (A) + Points (P)

FG% = FGM/FGA * 100
FT% = FTM/FTA * 100
P = Points = 2 * FGM + FTM
P/G = Points per game
D/Q = Defense per quarter
N/Q = Neutral per quarter
O/Q = Offense per quarter
+/Q = Plus per quarter = (D + N + O)/Q
-/Q = Minus per quarter = M/Q
T/Q = Total per quarter = (D = N + O - M)/Q

Team Totals for each item are calculated by taking the sum of all the players.

```
GET-$STIX
APP-$STIXII
RUN-180
STIX

* PLAYERS? 2
* DATA ITEMS? 16
OPTION? 1
NAME FILE? NONE
 1      ?3,2,3,3,5,2,1,1,2,3,4,12,0,0,3,1
 2      ?5,2,4,1,0,1,2,2,3,1,12,23,13,15,8,2
OPTION? 4
NAME FILE? NONE

                     BASKETBALL STATISTICS

 *      D1   D2   D3   D   N1   N2   N3   N    M1   M2   M3   M
 1       3    2    3   8    3    5    2  10     1    1    2   4
 2       5    2    4  11    1    0    1   2     2    2    3   7
TEAM     8    4    7  19    4    5    3  12     3    3    5  11

 *      A   FGM   FGA   FG%   FTM   FTA   FT%    P   P/G    0
 1      3    4    12   33.3    0     0    0.0    8   8.00  11
 2      1   12    23   52.2   13    15   86.7   37  18.50  38
TEAM    4   16    35   45.7   13    15   86.7   45  22.50  49

 * NAME    Q   G   D/Q   N/O   O/Q    -/Q    -/Q   T/Q
 1         3   1  2.67  3.33  3.67   9.67   1.33  8.33
 2         8   2  1.37  0.25  4.75   6.37   0.88  5.50
TEAM       8   2  2.37  1.50  6.12  10.00   1.37  8.62
OPTION? 10

DONE
```

Figure 4-20. Sample run of statistics program.

```
INVENTORY CODING

            FILE: INVENT - INFORMATION RETRIEVAL

    NAME        CATEGORY      USAGI    LOCATION           QTY.  YEAH  CONDITION

Article M P (10) 1 Baseball   1 MP     1 PE Storeroom     0-98  0-98  0 Shot
        W A      2 Softball   2 WP     2 A Storeroom                  1 Poor
                 3 Tennis     3 MWP    3 M Locker                     2 Fair
                 4 Golf       4 MA     4 W Locker                     3 Good
                 5 Basketball 5 WA     5 Coaches Office               4 New
                 6 Volleyball 6 MWP
                 7 Handball   7 REC
                 8 Hockey
                 9 Football
                10 Soccer                                      (continued)
```

Figure 4-21. Sample information retrieval program used in physical education.

INVENTORY

FILE; INVENT-INFORMATION RETRIEVAL

20 Character Name Maximum
6 Data Categories
ALPHABETIZED

NAME	CATEGORY	USAGE	LOCATION	QUANTITY	YEAR	CONDITION
Basketballs WP R	5	2	4	8	69	2
Basketballs WP L	5	2	1	2	72	4
Basketballs MP R	5	1	3	10	70	2
Basketballs MA L1	5	4	2	20	70	3
Basketballs MA L2	5	4	5	3	73	4
Basketballs WA L	5	5	4	24	72	3
Pucks	8	4	2	36	71	3
Soccer Balls REC	10	7	1	6	66	2
Soccer Balls MA	10	4	2	15	20	2
Basketballs REC R	5	7	1	7	63	0
Tennis Balls P	3	3	1	40	68	1
Tennis Balls A	3	6	2	25	71	2
Golf Balls REC	4	7	1	0	0	0
Golf Balls A	4	6	5	48	73	4
Softballs P R	2	3	1	16	67	2
Footballs MP R	9	1	1	12	66	2
Baseballs MA L1	1	4	2	30	71	2
Footballs MA R	9	4	2	10	69	3
Baseballs MA L2	1	4	5	24	73	4
Footballs MA L	9	4	5	6	73	4
Volleyballs REC R	6	7	1	9	65	2
Handballs REC R	7	7	5	6	60	3
Volleyballs WA L	6	5	4	12	71	3

(continued)

```
GET-$INFRET
RUN
INFRET

WHAT IS THE NAME OF YOUR INFORMATION FILE? INVENT

OPTION? SORT

SORT ON WHICH DATA NUMBER?5
0             GOLF BALLS REC
60            HANDBALLS REC R
63            BASKETBALLS REC R
65            VOLLEYBALLS REC R
66            SOCCER BALLS REC
66            FOOTBALLS MPE R
67            SOFTBALLS PE R
68            TENNIS BALLS PE
69            FOOTBALLS MA R
69            BASKETBALLS WPE R
70            BASKETBALLS MPE R
70            SOCCER BALLS MA
70            BASKETBALLS MA L1
71            TENNIS BALLS A
71            PUCKS
71            BASEBALLS MA L1
71            VOLLEYBALLS WA L
72            BASKETBALLS WPE L
72            BASKETBALLS WA L
73            BASEBALLS MA L2
73            GOLF BALLS A
73            FOOTBALLS MA L
73            BASKETBALLS MA 12

OPTION?STOP

DONE
```

Figure 4-21. Continued.

Information Retrieval

An information retrieval process may be used for the storing, accessing, and examining of large quantities of data, which can be of significant help in the physical education department where there are quantities of supplies, equipment, and data to keep track of. The sample inventory data file shown below illustrates how accurate records of supplies may be maintained. This kind of information retrieval may be done on an interactive basis using the same programs designed for storing and analyzing survey results in a social science class. The key to this use is coding the data so that they may be extracted in a meaningful manner

Simulations and Games

Athletic simulations are very popular with secondary students. The competitive factor and the accurate models used in many simulations provide a wealth of recreational activities which stimulate and challenge the users. The best activities for simulation are stop-action in nature—those with a natural pause in the pattern of play (such as golf, softball, or volleyball), at which time a decision may be entered. Various strategies of play may be tested with a simulation in an investigation for a best set of decisions. Figure 4-22 is a sample of an 18-hole golf game, of which only the first hole of play is shown.

```
GET-*GOLF
RUN
GOLF

WELCOME TO THE TIES TIMESHARING 18 HOLE CHAMPIONSHIP COURSE

TO GET A DESCRIPTION OF CLUBS, ETC. TYPE 0 FOR A CLUB NK.
WHEN REQUESTED

WHAT IS YOUR HANDICAP?9
OH-OH, A HOT SHOT!
DIFFICULTIES AT GOLF INCLUDE:
   0=HOOK, 1=SLICE, 2=POOR DISTANCE, 4=TRAP SHOTS, 5=PUTTING
WHICH IS YOUR WORST ?4

READY TO GO ?YES
YOU ARE AT TEE OFF HOLE 1    , DISTANCE 361   YARDS PAR 4

ON YOUR RIGHT IS ADJACENT FAIRWAY.
ON YOUR LEFT IS ADJACENT FAIRWAY.

WHAT CLUB DO YOU WANT ?0
HERE'S YOUR BAG OF CLUBS

  WOODS (FULL SWING ONLY)
     1    DRIVER
     2    BRASSIE
     3    SPOON

  IRONS (FULL SWING ONLY)
     12    TWO IRON
     .
     .
     19    NINE IRON

  IRONS (LESS THEN FULL SWING)
     22   TWO IRON - PARTIAL SWING
     .
     .
     29   NINE IRON - PARTIAL SWING

WHEN YOU REACH THE GREEN IT WILL BE ASSUMED THAT YOU ARE
USING A PUTTER.  THE PUTT POTENCY NO. REFERS TO THE STRENGTH
WITH WHICH THE BALL IS PUTTED.  USE NUMBERS GREATER THAN
ZERO, INCREASING THE NUMBER FOR GREATER DISTANCE.

YOU WILL BE ASKED FOR 'PERCENT FULL SWING' ON CLUBS 22-29.
THIS SHOULD BE A NUMBER FROM 1 TO 9.
--------------------------------------------------------------

WHAT CLUB DO YOU WANT ?1

SHOT WENT 251 YARDS - IS 110 YARDS FROM HOLE.
BALL IS 1   YARDS OFF LINE IN FAIRWAY.

WHAT CLUB DO YOU WANT ?19
TOO MUCH CLUB--YOU ARE PAST HOLE.
ON GREEN 60    FEET FROM PIN.  PUT POTENCY NUMBER ?7
PUTT SHORT.
ON GREEN 28    FEET FROM PIN.  PUT POTENCY NUMBER ?3
PUTT SHORT.
ON GREEN 15    FEET FROM PIN.  PUT POTENCY NUMBER ?3
PUTT SHORT.
ON GREEN 2     FEET FROM PIN.  PUT POTENCY NUMBER ?1
YOU H*O*L*E*D IT

YOUR SCORE ON HOLE 1      WAS 6
```

Figure 4-22. Sample computer golf game.

Computer-Managed Instruction

Many supportive curriculum computer applications classified in the general area of instructional management can play a significant role in the physical education curriculum. Two examples are considered.

Many schools participate in the National Youth Fitness Testing Program sponsored by AAHPER. A time-share computer program was written which stores the test results, assigns percentile ranks to them, computes a ranking mean as an index of overall fitness, and prints both a National Youth Fitness Report Card for each student and a teacher summary report for each class. Figure 4-23 shows a sample report printed by the computer.

Physical education teachers have been leaders in the individualized approach to developing requisite skills for effective participation in an activity. The problem of grouping students for work on particular skills is often time-consuming; however, the computer can offer the beleaguered teacher valuable assistance. Such a program, called ILA (Individualized Learning Activities), does exist. Objectives or skills

STUDENT NAME DOE JOHN SCHOOL K CLASS 7 STUDENT NO.

SEM.	SHUTTLE RUN SCORE	%	SIT UPS 2 MINUTES SCORE	%	PULL UPS SCORE	%	BROAD JUMP SCORE	%	50 YARD DASH SCORE	%	SOFTBALL THROW SCORE	%	600 YARD RUN SCORE	%
1	9.9	90	70	95	6	85	7.00	95	7.0	90	147	80	1.59	90
2	9.8	95	73	95	6	85	6.11	95	6.9	90	153	85	1.54	90

FIRST SEMESTER MEAN = 89
SECOND SEMESTER MEAN = 91

Figure 4–23. Sample computer-generated student report.

are identified and status on each for individual students is monitored and entered into the computer. The teacher may then have the computer locate students needing work on a particular skill and group them together.

```
GET-ILA
RUN
ILA

UNIT NAME?JOHNS

SØ = 52 S1 = 2 TØ = 11
>ROS
NO.      NAME                1   5   1Ø  15  2Ø
 1  AHRENS RICHARD          •••••    •    •   •
 2  ANDERSON MABEL              •    • ••••   •
 3  GREEN TIM                   •    ••••••••••••
 4  JOHNSON JAMES             ••••   •   • •  •
 5  JONES JONATHAN          ••••••••••    •   •
 6  MACK BONNIE             ••• •••• • ••••   •
 7  MILLER MIKE                 •    •    •   •
 8  RUTZ ELIZABETH             ••••  •    •   •
 9  SCHMIDT JENNIFER            •    •    •   •
1Ø  SMITH JOHN                  •    •   ••  •••
11  THOMPSON NORMAN           •••    •    •   •

>ADD-C,1,ON,14
>ADD-C,2,ON,6
>SIZ
GROUP  SIZE
  1     4
  2     4
  3     3
>LIS
GROUP 1                  GROUP 2                  GROUP 3
 2  ANDERSON MABEL        4  JOHNSON JAMES         1  AHRENS RICHARD
 3  GREEN TIM             5  JONES JONATHAN        7  MILLER MIKE
 6  MACK BONNIE           8  RUTZ ELIZABETH        9  SCHMIDT JENNIFER
1Ø  SMITH JOHN           11  THOMPSON NORMAN
```

Figure 4–24 Sample run of computer-managed individualized learning program.

SUMMARY

Current usage of the computer is minimal in physical education when compared to other disciplines; however, present trends indicate a substantial growth in computer use in a supportive role to assist the teacher in managing instructional activities. There is a need for programs and materials which use the computer effectively in the physical education curriculum, and the subsequent sharing of these ideas through articles and conference presentations.

NOTE: The programs used to illustrate various modes of computer usage were run on one of the Hewlett-Packard 2000C'/F Timeshare Systems at the Total Information for Educational System (TIES). TIES is a regional Minnesota cooperative network of thirty-four elementary, secondary and vocational-technical school districts providing data processing services for administrative, instructional, and research applications utilizing an on-line integrated data base.

COMPUTERS IN SECONDARY SCIENCE by James Friedland

INTRODUCTION

Science and its associated technologies are under fire for our pollution and energy problems. But recent surveys indicate that very few people feel we can solve our problems by abandoning science. For this reason, we will continue to require of students an understanding of science and its methods.

In junior high school, grades 7, 8, and 9, students are offered an introductory program, most often called general science. They study a wide variety of scientific fields and usually carry out simple laboratory exercises designed to introduce scientific procedures and tools. In addition, most secondary school students take science courses during their senior high school years, grades 10, 11, and 12. The most common courses offered are biology, chemistry, and physics. These programs examine a single field of science with its individual scientific approaches. High school science programs are almost always given with associated laboratories in which students carry out activities similar to those used by scientists in the field.

Science programs have recently passed through an era of revision. Traditional programs have been replaced by national "alphabet" curricula. The BSCS programs for biology, CHEM Study and CBA for chemistry, and PSSC for physics are but a few of the new programs guiding our instruction.

In this time of transition, two major trends can be discerned that have yet to be fully implemented in our schools. There has been a distinct change in the laboratory portions of science courses. Labs in the traditional programs were primarily observational activities. Students usually had enough information to predict their findings, and instructions told the student what he should be observing. Critics have characterized these as "cookbook labs."

Lab manuals for new revised curricula reveal a shift toward inquiring investigations. The labs are less directed and never tell the student what he should expect to see. Inquiry labs also involve measurement and data collection much more often than the traditional observational exercise.

The new curricula have been accompanied by a trend toward individualized instruction. Teachers are urged to tailor their instruction to each individual student rather than to the class as a whole. Science teachers usually find individualization difficult, since appropriate laboratory activities must be integrated with the normal classroom instruction.

Inquiry laboratories and individualized instruction, while easy to design on paper, are often extremely difficult to arrange in a secondary school environment. If the trends are to be fully implemented, our teachers must be allowed to make use of all the educational technology available. Such tools as 8 mm. filmloop projectors, overhead projectors, and filmstrip-magnetic tape self-learning modules will become increasingly important.

The computer can be regarded as an educational tool, in much the same light as a filmloop projector. Its tremendous flexibility, already demonstrated in business and research, offers many new educational strategies. We will look at several possible applications in this article, but by no means can we hope to outline all the possible uses.

USES OF THE COMPUTER IN SCIENCE INSTRUCTION

Drill and Practice

It is often true that a student in an introductory science program is expected to learn more new terms than in a foreign language course. Consider the drill-and-practice exercise about cell organelles in Figure

```
WHAT IS YOUR FIRST NAME?. JOHN

    YOUR ASSIGNMENT JOHN, IF YOU ACCEPT, IS TO CORRECT
CERTAIN FAULTS IN OUR MOST FAMOUS SCIENTESTS CELLS.

DOCTORS HAVE NOTICED A FAILURE IN HIS ENERGY PRODUCTION
SYSTEM.  TO WHICH CELL ORGANELLE WILL YOU DIRECT
YOUR MINI-SUBMARINE:

    1 RIBOSOME
    2 MITOCHONDRION
    3 GOLGI BODY
    4 CENTRIOLE

TYPE THE NUMBER OF YOUR CHOICE? 1

THE COUNCIL HAS DECIDED THAT YOUR DECISION WILL NOT SOLVE
OUR PROBLEM SINCE THE RIBOSOME IS THE SITE OF PROTEIN
SYNTHESIS.  PLEASE CHOOSE AGAIN.

TYPE THE NUMBER OF YOUR CHOICE? 2

VERY GOOD!  BUT WE HAVE A PROBLEM GETTING TO THE MITOCHONDRION.
WHICH OF THE FOLLOWING CELL ORGANELLES SHOULD PROVIDE
THE BEST PATHWAY TO REACH OUR GOAL?
```

Figure 4-25. Sample run of science drill-and-practice program.

4-25. There are many reasons for using the computer to teach such material. First, it never gets angry. It can respond positively to student errors, no matter how often a particular student is incorrect. One-to-one student-computer dialogue also makes it impossible for a student to hide behind the responses of classmates. Since the dialogue is private—with the computer rather than the teacher—computer drill and practice can also eliminate much of the anxiety associated with making mistakes, anxiety which can only hinder learning.

When a teacher assigns the class a problem set, every student receives exactly the same problems, making the likelihood of interstudent cooperation very high; and since the answers are covered in class at a later time, positive reinforcement for correct work is not immediate. A properly programmed computer can correct both these faults. Each student can be given a unique set of problems to solve. If the problems are generated by the computer, rather than read off a list, there is only a small chance of two students getting the same problems. Such a program can also respond immediately to a student's answer and offer another problem of the same type if the student should fail.

Evaluation of computer drill and practice with elementary school children shows that on the average, students can be expected to learn more but not as much as if they were individually drilled by the teacher. In secondary-school science the teacher usually has a difficult time working with a single student for an extended period of time.

Unfortunately, drill and practice requires a large amount of student-computer terminal time. In a classroom with a single terminal, drill and practice will be nearly impossible to schedule. But for a student having particular trouble with a subject, use of the computer in this fashion may be both effective and desirable.

Tutorial

Computer tutorial programs also require large amounts of computer time for each student. Studies in several universities put the cost at more than three times that of teacher-classroom instruction. Much work is going on in the hope of lowering this cost through application of modern technology.

Currently tutorial programs may be best used as part of a remedial program. Science teachers will find tutorial programs useful for students who have missed work and for students having trouble understanding the classroom instruction.

The areas where tutorial-remedial programs are needed can usually be identified by an experienced science teacher. Certainly most biology teachers would cite the Hardy-Weinberg Principle as a particular problem area. The program HARDY (Figure 4-26) includes instructional material designed to allow the student to apply the Hardy-Weinberg Law to a variety of problems.

As more computer time and terminals are made available to the science teacher, tutorial programs in many areas will be an immense aid in freeing the teacher to work with individual students, making individualized instruction a reality.

Problem Solving

The use of computers for problem solving will allow secondary science teachers to treat subjects in a quantitative fashion, when formerly the most that could be done was a look-and-see qualitative approach. The trend toward inquiry investigations makes this use attractive, since inquiry investigations often involve mathematical manipulations.

Sometimes, understanding the mathematical manipulations is not essential for appreciation of a

```
POPULATION GENETICS STUDY - THE HARDY-WEINBERG PRINCIPLE

HAVE YOU TRIED THIS PROGRAM BEFORE (1=YES, Ø=NO) ?Ø

     THE HARDY-WEINBERG PRINCIPLE PROVIDES A SIMPLE WAY
TO CALCULATE THE PROPORTION OF ORGANISMS IN A POPULATION
THAT ARE HOMOZYGOUS DOMINANT, HETEROZYGOUS DOMINANT, OR
HOMOZYGOUS RECESSIVE WITH REGARD TO A SPECIFIC TRAIT. ALL
WE NEED TO DETERMINE THESE PROPORTIONS ARE THE PROPORTION OF
ALELES IN THE GENE POOL WHICH ARE DOMINANT FOR THIS TRAIT
(P) AND THE PROPORTION WHICH ARE RECESSIVE (Q).

WE WILL GIVE YOU A POPULATION OF RATS TO WORK WITH. SOME
HAVE SHORT HAIR, SOME HAVE LONG HAIR. SHORT HAIR IS
DOMINANT, LONG HAIR RECESSIVE. THERE ARE 1ØØØ SPECIMENS.
HOW MANY RATS DO YOU WANT TO SAMPLE ?1ØØ

TOTAL NUMBER RECESSIVE = 15
TOTAL NUMBER DOMINANT = 85

WHAT PROPORTION OF THOSE EXAMINED SHOWED THE RECESSIVE TRAIT
THIS IS Q*Q ?15
PLEASE INPUT THE PROPORTION IN THE FORM OF A
DECIMAL NUMBER BETWEEN Ø.ØØ and 1.ØØ
WHAT PROPORTION OF THOSE EXAMINED SHOWED THE RECESSIVE TRAIT
THIS IS Q*Q ?.15

--------------------------------------------------------------

TO DETERMINE THE VALUE OF P AND Q AND THE GENOTYPIC
RATIOS, RECALL THAT:

1. P + Q = 1 SINCE THE PROPORTION OF ALLELES IN THE GENE
   POOL WHICH ARE DOMINANT FOR A TRAIT PLUS THE
   PROPORTION WHICH ARE RECESSIVE REPRESENT ALL ALLELES
   IN THE GENE POOL,

2. P*P REPRESENTS THE PROPORTION OF THE POPULATION THAT ARE
   HOMOZYGOUS DOMINANT,

3. 2*P*Q REPRESENTS THE PROPORTION THAT ARE HETEROZYGOUS
   DOMINANT,

4. Q*Q REPRESENTS THE PROPORTION OF THE POPULATION THAT
   SHOW THE RECESSIVE TRAIT, I.E. THAT ARE HOMOZYGOUS
   RECESSIVE,

5. P*P + 2*P*Q + Q*Q = 1 SINCE THE PROPORTION OF THE
   POPULATION THAT ARE HOMOZYGOUS DOMINANT, PLUS THE
   PROPORTION THAT ARE HETEROZYGOUS, PLUS THE
   PROPORTION THAT ARE HOMOZYGOUS RECESSIVE REPRESENT
   THE TOTAL POPULATION.

--------------------------------------------------------------

IN THE ABOVE EXAMPLE, Q = Ø.387298

GENOTYPE                    PROPORTION      NUMBER

HOMOZYGOUS DOMINANT            Ø.38            38
HETEROZYGOUS DOMINANT          Ø.47            47
HOMOZYGOUS RECESSIVE           Ø.15            15

DO YOU WISH TO REPEAT THIS PROGRAM (1 = YES, Ø = NO) ?Ø
```

Figure 4-26. Sample science tutorial.

scientific principle. This is the case for a BSCS investigation of population dynamics: Students are asked to examine how the population of birds varies on a hypothetical island given certain conditions. The following teacher-written program allows the student to quickly reach the important conclusions without applying complicated population growth formulas.

```
HOW MANY BIRDS ON ISLAND AT START? 6

CHANGE ASSUMPTION 1 (1 YES, Ø NO)? 1
HOW MANY OFFSPRING FROM EACH PAIR OF BIRDS? 8

CHANGE ASSUMPTION 2 (1 YES, Ø NO)? 1
WHAT PERCENT OF ADULTS WILL LIVE INTO THEIR 2ND SEASON? 15

CHANGE ASSUMPTION 3 (1 YES, Ø NO)? 1
WHAT PERCENTAGE OF OFFSPRING DIE BEFORE MATING? 5Ø

CHANGE ASSUMPTION 4 (1 YES, Ø NO)? 1
HOW MANY BIRDS ENTER THE AREA EACH YEAR? 5

NO. OF GENERATIONS? 1Ø
OUTPUT DESIRED (1 TABLE, 2 GRAPH, 3 BOTH)? 1

GEN.        TIME        POP.

Ø           Ø            6
1           1           18
2           2           43
3           3           94
4           4          2ØØ
5           5          422
6           6          882
7           7         1835
8           8         3824
9           9         7954
1Ø          1Ø       16527

ANOTHER RUN? (YES 1, NO Ø)? Ø
```

Figure 4-27. Sample science problem-solving program run.

Computers will also prove useful in situations where the math may be easy but the total solution of an investigation requires many repetitions of a simple equation. The following example involves a chemistry investigation that attempts to relate particle size to total surface area. After collecting the data, a student can find the total surface area by applying the simple formula:

$$\text{Surface Area} = 2 \times (\text{no. of pieces}) \times ((H \times L) + (H \times W) + (W \times L))$$

While this formula proves no problem, a full solution usually brings on student revolution, since it must be evaluated for up to 20 sets of data. The process often proves so distracting that the reason for doing the investigation becomes lost. The following computer program allows the student to concentrate on the relationship of particle size to total surface area.

```
SURFAC              17.53              12-MAR-74

HOW MANY PIECES ?1
LENGTH OF AVERAGE PIECE ?4Ø
HEIGHT OF AVERAGE PIECE ?2Ø
WIDTH OF AVERAGE PIECE ?2Ø

TOTAL SURFACE AREA  4ØØØ

NEXT

HOW MANY PIECES ?2
LENGTH OF AVERAGE PIECE ?2Ø
HEIGHT OF AVERAGE PIECE ?2Ø
WIDTH OF AVERAGE PIECE ?2Ø

TOTAL SURFACE AREA  48ØØ

NEXT

HOW MANY PIECES ?4
LENGTH OF AVERAGE PIECE ?1Ø
HEIGHT OF AVERAGE PIECE ?2Ø
WIDTH OF AVERAGE PIECE ?2Ø

TOTAL SURFACE AREA  64ØØ
```

Figure 4-28. Run of a science program which performs repetitious calculations.

Problem-solving programs are doubly attractive since the average amount of time that any student spends on the computer is usually quite small, allowing many students to use the computer in a single period. Most junior and senior high schools will be able to use the computer for scientific problem solving in the immediate future, at a reasonable cost.

Data Analysis

Large and complex data banks are still years away for classroom instructional use, since they require huge computers. But small-scale data banks are already in operation. In Scotland, local weather information stored on a computer is being used by secondary school students in science units on meteorology and ecology.

Simple data banks can be constructed using information from student investigations. This format will be particularly useful in studying long-term events, such as biological succession.

Computers, of course, will be used to statistically analyze the results of student investigation. In this role they will be functioning as problem solvers.

evaluating student results with formulae too complex to be processed by the students themselves. Statistical analysis programs suitable for high school science, such as chi square determination, are available from many sources, or can easily be programmed locally.

Most small data bank and data analysis applications can be successfully used with a minimum of computer time. In fact, direct student contact, while desirable, is not necessary. The data may be stored and retrieved or analyzed at the school's computer center at any convenient time, and be delivered to the classroom teacher.

Simulation

Simulation may be a new word to most science teachers, but it certainly is an old idea. In teaching situations when it is difficult to explore real phenomena science teachers have always resorted to replacement activities: chemistry teachers use models of atoms to explore chemical bonding, physics teachers use coiled springs to explore the behavior of electromagnetic waves, and biology teachers use coins to represent alternative alleles for genetics problems. All these activities are simulations, since one object is being used to represent another.

A brief look at two computer simulations being used in many secondary schools across the United States should illustrate some of the strengths of computer simulation in the science classroom. POLUT is used to study the effects of organic pollutants on bodies of water. Simulation is necessary, since most schools are not free to dump wastes into local streams and monitor the effects over several weeks. See Figure 4-29.

Alternate noncomputer simulations would prove very complex, since a reduction in scale would alter important parameters such as oxygen injection rates. An impossible laboratory becomes possible.

The Millikan oil droplet experiment is an important part of every physics course in high school, but the investigation is not carried out in lab. Teachers have found that the investigation calls for sophisticated equipment and great student manipulative skill. In this situation, a simulation would provide the student experience so effective in reinforcing a classroom concept. CHARGE, shown in Figure 4-30, is such a simulation.

Many computer projects around the world are producing simulations to complement secondary school science courses (see partial list of sources at the end of this article). Most of these simulations have been classroom tested and can be used with little modification on most school computers.

Thus, computer simulation is the most promising field for computer use. Science teachers have been devoted to the concept that the hand-on experience of laboratory work is essential for training future scientists and citizens living in the scientific world. Simulation provides a laboratory-type experience in situations where no lab is possible. The amount of student-computer terminal time varies with the simulation, but there has been a trend toward simulations that require only modest amounts of interaction, so that many students can work on a simulation during a single laboratory period. This makes many simulations possible for the teacher with access to only one computer terminal.

SUMMARY AND CONCLUSIONS

The coming of the computer to schools will not occur as rapidly as predicted by some of its most ardent spokesmen. Costs are still relatively high. Teachers are not being training to make use of the computer. And libraries of instructional materials are not yet easily available. But to ignore the instructional uses of computers would be a tragic waste of a valuable teaching device. This product of science is sure to be a major factor in our students' lives, as well as our own. We have only to look at the bills arriving in our daily mail to be reminded of this fact. There are uses for computers in our science classes, at reasonable cost. The applications involve either limited numbers of students, as in remedial work, or limited student-computer interaction.

Problem solving, data banks, data analysis, and simulations are the most promising uses for the school with limited computer facilities. Since they call for modest amounts of computer time, many students can take advantage of a single terminal. Problem solving programs are usually easy for a student or teacher programmer. Computer simulations will be written by computer development projects or commercial computer companies in such a way that the teacher will not have to be concerned with computer complexities.

While slow in coming, computers promise a most interesting future for science teaching.

```
                    WATER POLLUTION STUDY
INSTRUCTIONS (1=YES, 0=NO)?1

IN THI    DY YOU CAN SPECIFY THE FOLLOWING CHARACTERISTICS:

A. THE KIND OF BODY OF WATER:
   1. LARGE POND
   2. LARGE LAKE
   3. SLOW-MOVING RIVER
   4. FAST-MOVING RIVER

B. THE WATER TEMPERATURE IN DEGREES FAHRENHEIT.

C. THE KIND OF WASTE DUMPED INTO THE WATER:
   1. INDUSTRIAL
   2. SEWAGE

D. THE RATE OF DUMPING OF WASTE, IN PARTS PER MILLION (PPM)/DAY.

E. THE TYPE OF TREATMENT OF THE WASTE:
   0. NONE
   1. PRIMARY (SEDIMENTATION OR PASSAGE THROUGH FINE
               SCREENS TO REMOVE GROSS SOLIDS)
   2. SECONDARY (SAND FILTERS OR THE ACTIVATED SLUDGE
                 METHOD TO REMOVE DISSOLVED AND COLLOIDAL
                 ORGANIC MATTER)

***********

BODY OF WATER?2
WATER TEMPERATURE?35
KIND OF WASTE?2
DUMPING RATE?8
TYPE OF TREATMENT?0

DO YOU WANT: A GRAPH(1), A TABLE(2), OR BOTH(3)?1


AFTER DAY 4      THE FISH BEGIN TO DIE, BECAUSE
THE OXYGEN CONTENT OF THE WATER DROPPED BELOW 5 PPM.

            0...OXYGEN-SCALE....5...OXYGEN-SCALE...10...OXYGEN-SCALE...15
            0..WA  -.10..SCALE.20..WASTE.30..SCALE.40..WASTE.50..SCALE.60
DAY         I---------I---------I---------I---------I---------I---------I
 0         I   W                                             O
 1         I        W                                  O
 2         I           W                     O
 3         I            W             O
 4         I             W        O
 5         I             W     O
 6         I             W    O
 7         I             W    O
 8         I             W    O
 9         I             W   O
10         I             W   O
11         I             W   O
12         I             W   O
13         I             W   O

THE WASTE CONTENT AND OXYGEN CONTENT WILL REMAIN AT
THESE LEVELS UNTIL ONE OF THE VARIABLES CHANGES.

ANOTHER RUN (1=YES, 0=NO)?0
```

Figure 4-29. Sample science simulation, POLUT.

```
                MILLIKAN [illegible] EXPERIMENT

INSTRUCTIONS (1 YES, 0 NO) ?1

INSTRUCTIONS -- AFTER EACH QUESTION MARK, (V [illegible]), YOU MAY:

TYPE IN VOLTAGE BETWEEN -1000 and 1000 (IN ORDER TO MAKE THE
VELOCITY PRINTED OUT AS CLOSE TO ZERO AS POSSIBLE),
REQUEST CALCULATION OF CHARGE FOR STOPPED DROP (TYPE IN
2000), OR END THE PROGRAM (TYPE IN 3000)

                NO ELECTRIC FIELD

DROP:

 VELOCITY
(METERS/SEC)
[illegible]
                    [illegible]   [illegible]   29.8   29.9

V ? 200
                    -29.7   [illegible]   36.6   26.9
V ? [illegible] ?100
                    [illegible]   -7.2   [illegible]   -1.5
V ? 100 ?50
                    29.7   [illegible]   [illegible]   -7.2
V ? [illegible] ?150
                    -29.7   1.5   20   12.7
V ? 150 ?130
                    29.7   -0.2   [illegible]   7
V ? [illegible] ?2000

CALCULATION FOR WHICH DROP ?[illegible]

CHARGE OF DROP [illegible] IS [illegible] E-19 COULOMBS.
```

Figure 4-30. Sample run of science simulation, CHARGE.

REFERENCES

Sources on Instructional Uses of Computers

Books

Coulson, John E., *Programmed Learning and Computer-Based Instruction*. New York: John Wiley and Sons, 1962.

Holtzman, Wayne H., ed., *Computer-Assisted Instruction, Testing and Guidance*. New York: Harper and Row Publishers, 1970.

O.E.C.D., *Computer Sciences in Secondary Education*. Paris: Organisation for Economic Co-operation and Development OECD Publications, 1971.

Suppes, Patrick, *Teachers Handbook for CAI Courses*. Stanford University, California: ERIC document #ED054620, 1971.

Visich, Marion, and Ludwig Braun, *The Use of Computer Simulation in High School Curricula*. Stony Brook, N.Y.: Huntington Computer Project, State University of New York, 1974.

Articles

Abelson, P. H., "Fourth Revolution", *Science*, *177*, July 14, 1972.

Alpert, D., and D. L. Bitzer, "Advances in Computer-Based Education". *Science*, *167*, March 20, 1970.

Educational Technology Issue. *Saturday Review of Education*, *1*, May 1973.

Hammond, A. L., "Computer Assisted Instruction: Many efforts, mixed results". *Science*, *176*, June 2, 1972.

____, "Computer Assisted Instruction: Two major demonstrations". *Science*, *176*, June 9, 1972.

Schwarz, G., and others, "Computers in Physics Instruction," *Physics Today*, *23*, April 1970.

Suppes, Patrick, "The Uses of Computers in Education." *Scientific American*, (September 1966), 206.

Wahl, A. C., "Chemistry by Computer." *Scientific American*, *222*, April 1970.

Sources of Computer Programs and Program Ideas

Commercial Educational Sources

Digital Equipment Corporation
Educational [illegible]
Maynard, Massachusetts 01754

Hewlett-Packard Corporation
Educational Users Group Program Library
H-P Software Center
11000 Wolfe Road
Cupertino, California 95014

Time-Share Corporation
Box 683
Hanover, New Hampshire 03755

Non-Commercial Educational Sources and Projects

Chelsea Science Simulation Project
Chelsea College
Bridges Place
London S.W.6, England

Huntington II Computer Project
College of Engineering
State University of New York at Stony Brook
Stony Brook, New York 11790
(Huntington Computer Project materials are distributed through Digital Equipment Corporation—see above)

International Information Centre for Computing in Secondary Education
Moray House College of Education
Holyrood Road
Edinburg EH8 8AQ, Scotland

Physics Computer Development Project
Dr. Alfred Bork
University of California
Irvine, California 92664

Project LOCAL
Westwood Public Schools
44 School Street
Westwood, Massachusetts 02090

Project SOLO
University of Pittsburgh
Pittsburgh, Pennsylvania 15213

Scottish Education Department
Computers and the Schools, Final Report
Edinburgh, Scotland

Books and Articles

Davis, R. G., *Computer Programming in Quantitative Biology*. New York: Academic Press, 1971.

Orr, Marshall, Isenhour, and Jurs, *Introduction to Computer Programming for Biological Scientists*. Boston: Allyn and Bacon, 1973.

Zuckerman, David, and Robert E. Horn, *The Guide to Simulation Games*. New York: Western Publishing Company, 1970.

Luteri, G. F., and F. V. Scalzi, "Computer Program for Balancing Chemical Equations." *Chemistry*, *45*, November 1972.

COMPUTERS IN THE NATURAL SCIENCE CURRICULUM
by Herbert D. Peckham

INTRODUCTION

When the secondary curricula is viewed as a whole, the natural sciences must surely emerge as the part which contains the greatest number of obvious applications for the computer. This does not negate or detract from equally important applications in other areas, but merely recognizes that the stuff of science instruction interfaces naturally with the computer.

Natural science, as treated here, includes the subjects commonly found in the secondary curricula—physics, chemistry, and biology. Physics will receive the most attention, not because it is the most important but because of the author's background and the fact that more has been done here than in the other areas. The intent is to [illegible]rate applications which use the computer to [illegible]vantage. General methods or techniques that can be used across all three subject areas are developed.

Currently, the computer is used more in the problem-solving, or computational, mode than in any other; more and more educators, however, are beginning to see the importance of the computer with respect to games and simulations and in dialogues with students. The future will certainly see it employed in many of these new modes, in addition to that of problem solving. No matter which mode is involved, each application should be evaluated with respect to the following questions:

(a) What is the educational strategy involved?
(b) Is the objective a pedagogically sound one?
(c) Is the computer the most effective tool to use?

Far too often, a teacher can be dazzled into the use of the computer where it is completely inappropriate. The computer is a tool, albeit a very powerful one, which should be used only when it is the best method to obtain the desired results.

USES OF THE COMPUTER IN INSTRUCTION

There are many ways the computer has been used in instruction, and many new ways will be found in the future. It is therefore dangerous to establish hard-and-fast criteria or classifications, which are immediately argumentative and subject to challenge. For example, even acknowledged leaders in the field of computers cannot agree upon the precise meaning of CAI (computer assisted instruction) and CMI (computer managed instruction). Therefore, the classifications below should be taken not in a strict sense but only as a general guide. In many cases, the examples could be considered in more than one category. Some applications cross all classifications.

Drill and Practice

In this mode, the computer merely drills the student on material learned elsewhere. There are certainly instances in the sciences where the computer can be used in this way. Drill on the abbreviations for the names of the elements, or classification of plants and animals, would fall into this category. But is the computer being used to its best advantage here? Given limited facilities (which is nearly always the case), could any other method be used for drill that would free the computer for more important tasks?

An example will illustrate the point. One of the early Huntington project units was concerned with thin lenses. Most physics courses go into this topic, which involves the object distance, image distance, and focal lengths of a lens. If two lenses are combined; a standard sort of problem is obtained. Based upon given information, a relationship can be established between the object and the image. In this unit the computer gave the student a set of data with one element missing. The student was requested to compute the missing element and enter it into the computer, which then told the student whether he was right or wrong.

This is a prime example of how not to use a computer effectively. The value of the thin lens problem is not questioned; the point is that the computer is being used for such pedestrian application. Certainly the computer could generate large numbers of solutions to the problem, but these solutions can be presented to students easily and effectively without the computer. In my opinion, the computer as a drillmaster is the weakest mode of usage in the sciences. It does not take maximum advantage of the powerful capabilities of a very valuable educational resource. Accordingly, most of the attention in this article will be given to other modes.

Tutorial

The tutorial mode is at the same time the most promising, the most dangerous, and the most difficult. With imagination, it is possible to generate extremely effective tutorial units that bear little resemblance to the usual programmed text. Rather than identifying such units as tutorials, it is felt better to call them dialogs. In the best of all possible dialogs, it is as if a student and an instructor communicate in free form using computer terminals.

The fundamental difficulty in dialog generation involves the amount of time required to prepare and refine a unit. Estimates vary between 20 and 100 hours of preparation time per hour of polished dialog.

To illustrate what a very effective dialog in physics might look like, consider the following hypothetical conversation:

Computer: The task I would like you to perform is to compute how long it will take for an object to fall from the top of a building to the ground. What would you like to know?

Student: The answer.

Computer: Come on now, this should be a serious discussion. I am trying to get you to decide upon the information you need. What is this information?

Student: What is the mass of the object?

Computer: You really don't need this, but it is 1 kilogram.

Student: OK, I think that it will take 5 seconds to fall.

Computer: There seems to be a problem here. You have provided an answer without enough information to compute it. What additional information is needed?

Student: How high is the building?

Computer: The building is 20 meters high. Anything else needed?

(and so on.)

Note that the computer begins the conversation by drawing responses from the student without providing a great deal of information. Used like this, it is difficult to correctly analyze the student responses and respond reasonably to them. The computer must keep track of what information the student has requested so as to judge when a valid computation can be made. Each student who uses the dialog will produce an individualized set of responses based upon his input.

Very imaginative dialogs in physics, which include graphic displays of information, have been produced at the Physics Computer Development Project, University of California at Irvine. Most important to potential authors of dialogs in the secondary curricula is that effective techniques have been worked out to assist the author in preparing units. Dr. Alfred Bork of the Physics Computer Development Project has written a paper "The Computer in Learning--Advice To Dialog Writers," that is an invaluable aid to anyone serious about computer dialogs or tutorials. There are no large collections of tutorials or dialogs in the sciences at this time, but there is no question that properly designed computer tutorials can be of immense value in any curricula. The production of effective tutorials in science is an expensive, time-consuming, long-term enterprise that must be carefully planned if success is to be achieved.

Problem Solving

Most contemporary activity involving the computer in science instruction is in problem solving, and understandably so. There is a fundamental reason why the computer is so powerful to the secondary science curricula. Most natural processes or laws involve rates in one way or another; however, dealing with rates analytically requires calculus. This means that many valuable ideas must be put aside because of the mathematical difficulties involved. The computer permits the innovative teacher to go around these difficulties and examine nearly any topic desired.

As an example of this, Newton's Second Law states that $F = ma$. In this form, the student sees nothing but a simple lgebraic equality in three unknowns. Given two of these, the third can always be computed. Viewed this way, the student misses completely the rich content of the law, which describes rates of change of velocity and position. If the law is written i· a different form, the result is:

$$\frac{\Delta x}{\Delta t} = v \quad \text{and} \quad \frac{\Delta v}{\Delta t} = \frac{F}{m}$$

In these equations the symbol Δ is read as "the change in," for example, "the chang ı x divided by the change in time equals the velocity." A little algebra is all that is needed to put the equations into the final form below:

$$x_{new} = x_{old} + v_{old}\,\Delta t$$
$$x_{new} = v_{old} + \frac{F}{m}\Delta t$$

Now what is usuall· oblem involving a second-order differ as been converted to a simple algebr. on that is easily motivated and can be use . ents who have had only the rawest elements of algebra. At a very fundamental level, the goal of mechanics is to predict the motion of an object given its position, velocity, mass, and the forces that act upon the object. But this is precisely what is available in the equations above! The same set of equations is valid in situations where calculus based methods fail.

The pocket calculator can be used to great advantage in science instruction. This is illustrated with an example that uses the equations of motion given above. Suppose that a mass m is driven by a spring with spring constant k giving a force of $-kx$. If $k = m = 1$, then the following set of e ations is obtained:

> For the first velocity calculation a half step in time is used to compute the velocity midway in a time interval.

$$v_{new} = v_{old} - x_{old}\,\Delta t/2$$

Thereafter the equations are:

$$x_{new} = x_{old} + v_{old}\,\Delta t$$
$$v_{new} = v_{old} - x_{old}\,\Delta t$$

The strategy of one velocity computation with half a time step is called the "half step method" and produces a dramatic increase in accuracy. If a time step of 0.5 is used, and if when $t = 0$, $v = 0$ and $x = 1$, any pocket calculator can be used to obtain the results below very rapidly.

t	v	x
0.0	0.000	1.000
0.5	-0.250	0.875
1.0	-0.687	0.531
1.5	-0.952	0.055
2.0	-0.979	-0.434
2.5	-0.762	-0.815
3.0	-0.354	-0.992
3.5	0.142	-0.921
4.0	0.602	-0.620
4.5	0.912	-0.164
5.0	0.994	0.333
5.5	0.827	0.746
6.0	0.454	0.973
6.5	-0.032	0.957

If x is plotted against time, a very nice cosine function is obtained which is, of course, the result obtained using calculus. The point here is that a computer method converts differential equations into algebraic equations which may or may not require the computer! As illustrated above, a pocket calculator can produce dramatic results. The most important idea might be summed up in the principle that "the more you use a computer, the less you need to use the computer." Once students become familiar with computer methods, an idea can be explored simply by discussing how the solution would be obtained. Many times this provides the desired insight into the problem.

The data above illustrate a fundamental problem with computers in the problem-solving mode. The output is usually in the form of columns of numbers. It is very difficult to acquire a feel for a solution using this kind of output. Consequently, if at all possible, the results should be output to a graphic display device of some kind—an XY plotter or a graphic terminal. Crude graphics can be accomplished with the [illegible]. A good deal of it is done in the curri- [illegible] already mentioned. One of the [illegible] programs (PLOTTR) provides a [illegible] capability on the teletypewriter. This [illegible] however, only when the function is defined. Many times a solution of one of the rate problems discussed above does not have a closed form solution, and other techniques must be used. Several new hard-copy terminals on the market based on the Diablo printing mechanism can produce excellent graphic output equivalent to an XY plotter. If no graphic display devices are available, students can be taught to quickly scan a column of numbers for such characteristics as maxima, minima, and zero points. Crude graphs can be sketched quickly with this information. As the price of graphic display devices drops, careful consideration should be given to the pedagogical value of graphic output.

A good deal of excellent curriculum material in the problem-solving mode is currently available to the science teacher. Some of the Huntington Project programs in this area are listed below.

EVOLU	A natural selection experiment.
NZYME	Enzymatic reaction rates.
PHOSYN	Photosynthesis rate experiment.
REFLCT	Reflection through the least time principle.
SPACE	Characteristics of orbits (nongraphical).
WAVES	Sum of two waves.
MOLAR	Acid base titration.
PHPOH	pH and pOH percent dissociation for weak monoprotic acids.
DECAY2	Nuclear decay.
EQUIL1	Equilibrium of $2HI \rightleftharpoons H_2 + I_2$.
EQUIL2	Equilibrium of $PCl_5 \rightleftharpoons PCl_3 + Cl_2$.
KINET	Reaction rate of $A \rightleftharpoons B$.
MASSD	Mass defect.
BFIELD	Magnetic field picture.

The Hewlett-Packard Curriculum Project has complete instructional units in science on Geometric Optics, Mechanics, Electricity and Magnetism, and Waves. All these are suitable for secondary school use.

The list of applications above is a representative sample of a larger collection. The instructor who starts using units such as these will most likely begin to develop his own applications as time goes on. The differences in classes and in school systems makes this highly desirable.

Data Analysis

Sooner or later, all science students become involved with acquiring and processing experimental data. The computer has an obvious and valuable function in this process. Simple programs to compute the mean and standard deviation of a set of data are very easy to write and often are valuable student exercises. One of the Huntington Project programs, STAT, performs statistical analysis of laboratory data. Most of the program libraries furnished with computers contain very sophisticated statistical analysis programs. (It is questionable whether these types of programs are of much value in the secondary curricula, since they presuppose a knowledge of mathematical statistics.)

Very little more needs to be said with regard to the computer (or pocket calculator) and data analysis in the science laboratory. It is, however, important to note that the computer is not being used in a particularly imaginative mode here. It is regrettable that faculty often feel this is the only place computers can be used in science instruction. Data analysis is certainly important, and computers have an important application here, but not to the exclusion of other applications that are probably more valuable.

Simulations and Games

Computer games provide a vast potential waiting to be exploited by the wise educator. It is the common experience at all computer facilities that students love to play games on computers. Incredibly involved and detailed games have been developed. The "Peoples' Computer Company" has a list of entertaining games for which program listings are available. "101 BASIC Computer Games," the first known collection of games solely in the BASIC language, is available from the Digital Equipment Corporation.

With all computer games, it is important to decide whether the computer is providing merely relaxation or educational instruction. The best games are those which involve meaningful concepts difficult to teach in other ways. A particularly good example is "Fermatopolis 500," conceived by Murray Alexander of De Anza College. In this game, two drivers start at point A and the race finishes at point B. It is an unusual type of race because the drivers cannot control the speed. The computer assigns the same velocity to each car in region I and a different velocity to each in region II.

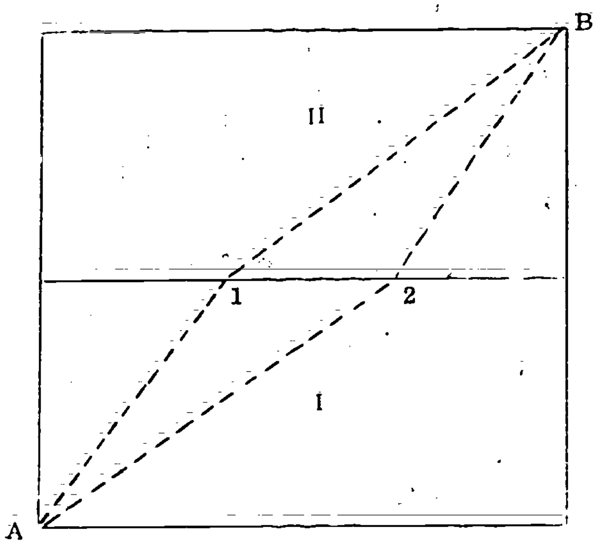


Figure 4-31. Diagram of a race in the game "Fermatopolis 500".

The only control the driver has is to choose where he wants to cross from region I into region II. The objective is to reach B in the minimum time. A graphic version has been implemented in the Physics Computer Development Project. The game is designed to teach Fermat's Principle—one of a class of important optimization principles in physics. Normally, these ideas are investigated using calculus, but they can be profitably explored in the computer game format.

Simulations

In the long run, computer simulations may well turn out to provide the greatest educational gain for the largest number of students. A depressingly large percentage of science classes are forced to treat subjects peripherally, never coming to grips with the central issues. This occurs because of the almost complete lack of mathematical skills of most students. With a minimum of beginning algebra, though, students can be taught to construct computer models which permit them to acquire valuable insights into what are often very involved problems not susceptible to mathematical analysis. "Air Pollution," one of the Hewlett-Packard Curriculum units, leads students through this type of process. A second unit in the series, which requires no programming, is an ecological simulation called GRAZE. The following simulations from the Huntington Project have been expanded by the Digital Equipment Corporation and have back-up material available:

STERL Fly Population control.
GENE Genetics simulation.
POLUT Water pollution simulation.

An excellent reference for modeling of processes is *The Man-Made World*, published by McGraw Hill.

SUMMARY AND CONCLUSIONS

There is no question that the computer can be of vast importance to the science curriculum. The surface has just been scratched. So far, the computer has been used generally to do what was going to be done in any case, though possibly in a more imaginative and innovative mode. What needs to be done at all levels is to examine critically what one would like to do in a given curriculum, quite apart from existing texts and reference materials. This process is difficult and frustrating, but is very valuable. The computer should be viewed as a tool which is available to assist the educator. A course should be quite different, both externally and internally, depending upon whether a computer is being used or not; in fact, this criterion is a good one to judge the effectiveness with which the computer is being utilized in a specific course.

A final point is most important. The yeast of effective computer utilization in the science curriculum is a collection of good ideas. Things that seem obvious to one faculty member and not worth repeating may not be at all obvious to another. Collective, organized action by a group of qualified and interested educators is much more likely to produce high-quality results than the efforts of isolated individuals. This is particularly true of computer tutorials or dialogs. The time required to produce and test effective units precludes the generation of significant collections by an individual; the group effort is mandatory.

The computer has exploded into the educational community with a speed that could not have been predicted ten years ago. The gauntlet has been thrown down. The potential gain is surely greater in the science curriculum than any other. How well education meets the challenge remains to be seen.

SOURCES

Physics Computer Development Project, Department of Physics, University of California, Irvine, California 92664.

Hewlett-Packard Company, 11000 W[illegible] Road, Cupertino, California 95014.

Digital Equipment Corporation, 146 Main Street, Maynard, Massachusetts 01754.

People's Computer Company, P.O. Box 310, Menlo Park, Califor[illegible]025.

COMPUTERS IN THE SOCIAL SCIENCES by Allen D. Glenn

INTRODUCTION

Social studies instruction in today's classrooms reflects a kaleidoscope of historical precedents and innovative trends in instructional strategies and materials. Many courses are dominated by the historical approach to the study of man and by the lecture method of teaching. An increasing number of courses, however, reflects the newer and more innovative teaching strategies and materials that were developed in the late sixties and early seventies. In these classrooms the analysis of various topics, issues, and questions is undertaken through a social science perspective with emphasis on the scientific method of inquiry and the exploration of various values and value questions.

The social studies teacher of the seventies is charged with presenting a realistic picture of man's world and the problems he faces, and with teaching students the analytical skills necessary to enable them to confront the problems and demands of an ever-changing environment. Social studies teachers who take this charge seriously rely on a variety of instructional strategies and seek to meet the individual learning needs of each student. As a consequence, three major elements characterize social studies education:

1. Emphasis on the social science disciplines as a basis of study.
2. A focus on analyzing the individual and collective value positions as they relate to various value questions confronting man.
3. Increased emphasis on a more individualized, flexible approach to social studies.

Computer-assisted instruction is in the forefront of the newer instructional strategies being used in the social studies classroom. Until just a few years ago the computer was viewed by most social studies teachers as a thing that belonged to the math classroom; recently, however, more and more teachers have begun to realize the flexibility and potential of the computer as it relates to the study of classroom topics, the analysis of data, and the exploration of value questions.

During the next few years, computer-assisted instruction will become an integral component of a social studies classroom's learning environment. A major factor in this growth will be the increased number of computer facilities and equipment that will be made available. Also, additional materials related specifically to social studies instruction are being developed and distributed. The most crucial factor in this increased use of the computer, however, is that the social studies teacher will have opportunities to become versed in how the computer may be used as an instructional tool in the classroom.

USES OF THE COMPUTER IN INSTRUCTION

It is a mistake to consider the computer as a new instructional gimmick to include in a teacher's bag of tricks. Such a conception causes one to miss the tremendous potential of the computer as an instructional tool to assist teacher and student in achieving the instructional objectives of a unit of study. The computer and its related materials must be considered as an integral part of the total instructional process. The teacher must plan ahead for the most effective way to use the computer in meeting the goals of the unit and the individual needs and learning styles of the students.

Drillmaster and Tutor

Learning facts, concepts, and generalizations has always been an integral part of the social studies. In fact, some critics suggest far too much of social studies instruction focuses on memorizing facts and figures. Whether one agrees or disagrees, there are facts, concepts, and generalizations that are important antecedents to higher level learning.

Drill and tutoring programs can be used effectively in the social studies classroom in two ways:

1. They may be used as an instructional strategy to assist students in learning specific facts. For example, in an American history class in which there are many important facts, the teacher may develop a drill or tutoring program to help the students learn them. During the unit each student, depending on his own needs, may call up the program and check his learning. This enables him to learn at his own pace and lets the teacher and class explore other areas of importance to the unit.
2. They may be used as a part of an individualized learning packet. More and more social studies teachers are either using these as part of their program or having students develop grading contracts. A drill or tutoring program included in the student's materials or contract would ensure that all students are acquiring the basic elements of each unit.

These two types of activity let the student learn at his own rate, check his own learning progress, and determine what areas he may need help in. They also release the teacher for other instructional activities.

While drill and tutorial programs are valuable as learning activities and offer significant assistance to the teacher, they do not utilize the full potential of the computer or meet the needs of most social studies curricula. In my opinion, it is in the areas of problem solving, data analysis, and exploration of values that computer-assisted instructional programs are most valuable to social studies programs.

Problem Solving and Simulation

An integral component of the social studies program is teaching the scientific method of inquiry as a means to problem solving. Students are to learn how to speculate about possible answers to a question, develop hypotheses, collect relevant data, test their hypotheses, and draw tentative conclusions. It is during this important phase of social studies instruction that the computer can be most effective as an instructional tool. At present, a variety of packaged computer programs and simulations are available for use in the classroom. Three excellent examples are USPOP, POLUT, and ELECT 1,2, and 3, developed by the Huntington Two Project. These computer simulations permit the student to manipulate a variety of variables and can be used to test hypotheses. They offer opportunities for interaction between the speed and flexibility of the computer and the supporting classroom materials and instructional activities. USPOP illustrates the potential of such programs.

USPOP is a human population model oriented toward investigating United States population projections. The student can investigate the effects of fertility, age of mother at birth of child, sex ratio of offspring, and age-dependent mortality on population size and structure. The program can be used in a classroom in which students are studying topics relating to the growth of United States population—a government or civics class, a sociology class, a current problems course, a geography course, or an American history class. In any of these, students could begin to explore questions concerning population growth. In a geography class studying population density, for example, students might want to explore what effect zero population growth will have on the population by 2050 and then speculate about the changes that might occur in densely populated areas. By determining the growth rate and controlling other variables, students could develop hypotheses and test them by interacting with the program. Because the program provides instant feedback, the students could see immediately the results of their speculations and apply the data to their original questions about population density. By additional manipulation other variables might be controlled and students could explore a variety of questions.

Such programs as USPOP offer the social studies teacher a variety of instructional opportunities. A particular program may be incorporated into a unit and used as a classroom exercise in which the class discusses and explores selected topics with the teacher controlling the questions and areas of study. Such a program would also be useful in teaching students to develop and test hypotheses and draw conclusions. Here, the computer program would be a skill-building exercise. It may also be used to let students explore questions that are related to the topic of study but go beyond what is to be considered during class discussion. Small groups of students might explore specific questions, or an individual student might seek answers to a question of special concern to him. Still another use of problem-solving programs would be in the area of evaluation. For example, after the class has studied the growth and determinants of population, the teacher might run a specific set of variables through USPOP and ask the students to discuss the results. Such an activity would be an excellent summative evaluational technique.

Value Questions

All of the programs mentioned above permit the student to explore a variety of "what-if-we-did-this" inquiries. Such questions offer the student and teacher an opportunity to explore numerous value questions related to a specific topic. For example, many students raise questions about the most effective way to control population. Such alternatives as sterilization and forced birth control are offered. Computer programs like USPOP not only let the student observe the impact of such actions on the population, but also permit exploration of the question, "Should we do this act, given the possible impact on our population?" The possible answers raise issues and concerns that are a vital part of today's social studies curriculum. Because students can observe the impact of their decisions, they may think more seriously about their own values and decisions.

Data Analysis

Analyzing and collecting data are skills related to problem solving. Students must be able to manipulate data that have been collected and analyze the results of the statistical analysis. Data banks (collections of information about a selected topic) are currently being used in social studies programs from elementary school to senior high school. The manual process of analyzing the d[illegible]uming task; computer-based program[illegible] feedback and flexibility can be of [illegible]: With minimal effort students can selec[illegible]mation from the data bank, complete selected analysis programs, and begin to develop tentative conclusions.

Some data-analysis programs also let students enter and analyze their own data. In many social studies classes students develop questionnaires, draw samples, and collect data. For example, American government students collect data about people's attitudes toward the government. With the assistance of computer packages developed specifically for data analysis purposes, students may create their own data bank and select the appropriate variables and statistical techniques. These programs increase student involvement in the analysis process and increase his learning about what it means to collect and analyze data related to a specific problem.

CONCLUSIONS

Computer-assisted instruction may be of tremendous assistance to the social studies teacher. The computer offers a means of improving and enhancing instruction, and its flexibility can provide an additional "teacher" and an open laboratory for exploring and analyzing a variety of topics. Used effectively, the computer can widen the horizons of social studies instruction and more effectively meet the needs of today's young adults. But it is the teacher who holds the key to the use of the computer as an instructional tool; not until he decides to incorporate it into the total learning and instructional process will the computer be more than another gimmick in a bag of tricks.

SOURCES

Information on computer programs for the social studies may be obtained from the following sources:

Software Distribution Center
Digital Equipment Corporation
Maynard, Massachusetts 01754

Digital produces all materials developed by Huntington Two Project. Huntington II has produced a variety of social science packages for use in the classroom.

Tecnica Education Corporation
1864 S. State Street
Salt Lake City, Utah 84115

Conduit
P.O. Box 388
Iowa City, Iowa 52240

This is a clearinghouse for computer assisted instruction.

Teachers should also check with the social studies coordinator within their district to determine what is available at the present time.

COMPUTERS IN THE SOCIAL STUDIES by Joseph U. McGrath

INTRODUCTION

The new social studies emerged in the early 1960's. While probably no single element of it is entirely new, the changes that have occurred over the last decade in teaching high school social studies have been extensive. Broadly stated, these reforms involved moving in the direction of

1. Greater emphasis on the social sciences;
2. Greater stress on the methodologies of research or modes of inquiry;
3. Greater concern with contemporary problems and world perspectives;
4. Greater interest in attitudes and values, the affective domain;
5. Greater use of a variety of materials rather than the single textbook;
6. Greater utilization of varied teaching methods, especially those emphasizing student initiative.

In brief, the traditional content and techniques of teaching social studies have been greatly expanded and updated. While the reforms have not been uniformly successful, it has been an invigorating decade professionally for the experienced social studies teacher.

It now appears that we are in a period of consolidation. Many institutes and projects are closing down. The professional literature seems far less radical than before. Private publishers have taken over the publication and distribution of materials that were originally produced by federally funded social studies projects. These materials are finding their way into an ever greater number of classrooms. So, while the creative stage of the new social studies appears to have passed, we shall certainly never go back to the old social studies, despite the current nostalgia for the 1950's; change and evaluation of materials and techniques are still very much a part of the typical high school social studies classroom. It is in this context that the computer will be used in social studies education.

The computer is an event in human history. In its impact on the life of man, it seems destined to rank with printing, radio, and television. It is part of our world and even more part of the future world of our students. As such, it clearly belongs in the social studies curriculum. We have a responsibility to teach technological literacy. Our students cannot understand contemporary American government, business, or industry without knowing what a computer is and what it can and cannot do.

One of the basic principles of the new social studies—one that goes back at least to John Dewey—is that students learn through experiencing a process. While we certainly have a responsibility to teach about the computer as an object of instruction, the most valuable learning will occur where we use the computer as a tool for instruction; it is in this use the student can be given that hands-on experience which is such an important element in achieving real understanding. Used in its several roles, the computer can contribute to social studies instruction in a variety of ways.

But social studies teachers don't know anything about computers! This may be true for many of us. Not only do we not know much about computers, we are intimidated—even frightened—by them. Despite their historic dimension, computers are really not our thing. Rarely, if ever, have they come into the high school in response to the demands of the social studies department. Mathematics and science departments and the school administration have provided the leadership in making computers available to the schools. Ours is a humbler task. We have access to a computer, or will have in the near future: how can we use this opportunity to enrich our teaching and fulfill our obligations to our students?

USES OF THE COMPUTER IN SOCIAL STUDIES INSTRUCTION

Drill and Practice

Functioning as a drillmaster in the social studies classroom, the computer can relieve the teacher of that thankless job, save his time, and provide the student with an interesting way to master some of the basic facts and concepts of history and the social sciences.

Computer-conducted drill has several obvious advantages over a question-and-answer drill session with an entire class. The student can take time to think out the answers without always having another student answer first, so that each student progresses at his own rate. The fact that the student learns immediately whether he is right or wrong and is then given the exact answer is a decided advantage over the usual written exercise. Finally, at the end of the computer exercise the teacher has a record of the student's performance.

Although drill is not looked upon with favor by most exponents of the new social studies, many classroom teachers find it a valuable teaching aide when used selectively. The drill programs are usually simple and can be constructed by the teacher with the help of a student who knows BASIC programming.

Tutorial

For the social studies teacher the most valuable tutorial programs are those which present and explain a concept. Has any high school American history teacher developed a technique for presenting the money question of the 1890's that did not leave some students bored and others confused? How efficient it would be to be able to send the students to a computerized tutorial program that would allow each to learn at his own pace. This function of the computer will become increasingly relevant as we integrate more of the concepts from the social sciences into our social studies curriculum.

Problem Solving

Marshall McLuhan has pointed out that computers are extensions of our brains, as tools are extensions of our bodies. This is a helpful concept to keep in mind when considering the use of the computer as a problem solver. Certainly problem solving has long been a central theme in social studies education.

In a problem-oriented social studies curriculum, the computer can be a most effective tool in allowing the students to deal with information at a level of complexity and sophistication that has not previously been possible. Applications in population studies, ecology, and economics immediately come to mind. A typical problem might be:

> If in 1978 Congress passed a law limiting families to two children, what would the population of the United States be in the year 2000?

The answer to such questions, quickly provided by a computer, can be the springboard to interesting and instructional analyses, papers, and discussions of many sorts. The validity of the program model determines in part the accuracy of the solutions the program generates. This is a most important concept for the social science student, as it concerns not only the validity of computer results but the validity of any model-based analysis. In this context, recognition that the quality of the data put into the computer determines the quality of information coming out ("garbage in, garbage out") is important and relatively easy to get across to students. In addition, use of the computer for problem solving naturally invites discussion of how the computer operates as well as what it outputs—important topics for students.

It is especially in the area of using the computer as a problem-solving tool that attitudes toward the computer can and should be developed. It is, after all, a tool that is used as the men who control it want it used. If it is being used in such a way as to threaten our privacy or freedom, our complaint is against those who are so using it. In the wrong hands, the computer can pose a threat to freedom. But that threat can clearly be contained by a concerned and knowledgeable citizenry within our democratic traditions.

Data Analysis

A number of elements of the new social studies support the educational value of having students investigate problems through the collection and analysis of data. For example, the *High School Political Science Curriculum Project* recommended the use of the high school itself as a laboratory for testing propositions. Here the student social scientists use their school and their classmates as subjects to be surveyed.

In school or out, the student research paper based upon a survey or other sources of quantitative data is increasingly common. While the data are neatly presented in these student research papers, typically nothing is done with them beyond a simple quantitative comparison. The computer allows a much more sophisticated analysis and evaluation. It also encourages the students to expand both the amount and the type of information they secure.

Another educational use of the computer in data analysis involves allowing the student to test hypotheses by using a comprehensive collection of data that is part of the program. The Educational Systems Research Project at the Carnegie-Mellon University has a system--the U.S. Congress 1829-1836--*which provides the student with data on 936 individuals who served in Congress during this period. Each congressman is described by fifty-two categories of information. Also included is data on 5,000 Congressional bills or motions that were introduced and how each congressman voted. Here certainly are data to test all those theories of who supported Andrew Jackson and why. It also should give the student an appreciation of the complexity of the past. But, most important, it should give him an understanding of what it means to formulate a hypothesis, test it, revise it, test it again, and so on.

Simulations

Computerized simulations and games provide the classroom teacher with one of the most inventive and promising instructional innovations. For social studies, simulation provides a self-contained environment in which the student or the class can study the cause-and-effect relations in a model of a complex real-life situation or system. Here the student social scientist can experiment in economics, political science, or sociology as his counterpart in the science laboratories can experiment in biology, chemistry, or physics. Using the computer, the student can explore the model and formulate, test, and revise hypotheses. In the Huntington Two simulation MASPAR, for example, the student is given the opportunity to explore a model of the relationship between class structure and political participation, among other variables. He then is able to do something that no political scientist can do in the real world: he can change the class structure simply by typing in different numbers and watch what happens to political participation.

Educational game programs are also based upon simulations of real-life situations or systems and are highly motivational to students. MARKET, another Huntington Two simulation, illustrates the game dimension of a simulation of economic competition. Two groups of students assume the roles of presidents and other officers of two companies manufacturing and selling bicycles. Each team of executives makes decisions about production, advertising, and pricing for their company during the forthcoming quarter. After the decisions for both companies are entered into the computer, a financial statement for each company is printed out. The two teams make their decisions for the next quarter on the basis of these financial reports. The game ends when one company accumulates 12 million dollars in total assets or goes bankrupt.

Even in situations where the one computer terminal is located in the math department office, it can be used as an instructional tool in the social studies curriculum. For example, POLICY is one of a number of computerized simulation games that can be played without direct contact with the computer. It is a simulation of the interest group process in formulating national policy. The students are members of one of six interest groups: Business, Labor, Civil Rights, Military, Internationalists, and Nationalists. Each student is given a list of socioeconomic indicators that portray the health of the country and a list of policies which if passed could change the indicators. Each interest group compares its goals with the national indicators, examines the policies, and decides which to support. Realizing that they need the support of other interest groups, they bargain to gain the votes for their policies. At the end of the class period each interest group turns in a form telling which policies it supported with its 100 influence points.

Any time later in the day, the teacher or a student can enter these data into the POLICY program. After the computer determines which policies have passed, it prints out the resulting changes in the socioeconomic indicators. The teacher can duplicate this computer printout and hand it out to the students at the beginning of the next period, when it becomes the basis for the second day's deliberation. The class is thus able to benefit from computer simulation without ever having direct access to the computer.

*Social Education (December 1970), p. 905.

SUMMARY AND CONCLUSIONS

The computer represents a promise and a challenge to the social studies teacher. The promise is evident in everything that has been discussed. This fantastic tool can extend and enrich student learning beyond belief. The challenge is to adopt it and adapt it to our teaching methods and educational goals. Certainly the modes of instructional use as described above are compatible with the thrust of the new social studies. In our discussion of typical uses at least one computer application was mentioned for each of the six reforms listed in the introduction. And it must be emphasized that computer-based instruction is being proposed as a complement to existing educational techniques.

School time, however, is limited. New methods must not only be educationally sound, they must be more efficient than those they replace; and they must replace something. What then are the unique advantages of teaching with computerized instruction?

1. The students are more highly motivated because they are more actively involved.
2. They learn the skills of procedure, analysis, and decision-making.
3. They learn the relation of factual information to concepts and how relevant information is used to test hypotheses.
4. They increase their self-understanding as they see how they act in different learning modes.
5. They probably gain an improved sense of their own learning abilities when they are allowed to proceed at their own pace.
6. Student-teacher relations improve as they are less frequently in adversary positions.
7. Participating in computerized classroom simulation games results in a more open classroom setting and a greater independence of students.
8. More students are more actively involved in class activities.
9. Attitudes and values change as students come to understand, appreciate, and empathize with the roles they play in simulated experiences.
10. They get to know their fellow students better through interacting with them in the games.

The promise is there; the challenge is yours.*

BIBLIOGRAPHY

Braun, Ludwig, "Digital Simulation in Education." *Educational Technology Systems* (June, 1972), 5–29.

Cardenas, Alfonso, and Leon Presser, (eds.), *Computer Science*. New York: Wiley-Interscience, 1972.

Collins, N. L., and Donald Michie, (eds.,) *Machine Intelligence*. New York: American Elsevier Publishing Co., 1967.

Education Products Group, *EDU*. Digital Equipment Corporation, Maryland.

Heitzmann, William Ray, "The Validity of Social Science Simulations: A Review of Research Findings." *Education* (November–December 1973), 170–175.

Inbar, Michael, and Clarice S. Stoll, *Simulation and Gaming in Social Science*. New York: Macmillan, 1972.

Koch, Warren, *The Use of Computers in Instruction in Secondary Schools*. Washington, D.C.: The National Association of Secondary School Principals, 1972.

Margolin, John B., and Marion R. Misch, (eds.) *Computers in the Classroom*. New York: Spartan Books, 1970.

Martorella, Peter H., and Dixie Kohn, "Computer Related Materials in the Social Studies/Social Sciences." *Social Education* (December 1970), 899–908.

McLuhan, Marshall, *Counter Blast*. New York: Harcourt, Brace, 1969.

Simulation and Games. Beverly Hills: Sage Publications,

*The computer programs mentioned are available from Digital Equipment Corp., Maynard, Mass.

COMPUTERS IN VOCATIONAL AGRICULTURE
by Jerry Paulson

INTRODUCTION

Agriculture today is less dependent on manpower than it has ever been. Less than 10 percent of the workers in the United States make their living on farms, as contrasted with about 60 percent in 1850. In order to succeed, the farmer must be a skilled manager: he needs to make decisions not only about the cultural practices required to produce crops or animals, but also about capital investments, marketing strategies, production practices, and the like.

In the curriculum of agriculture as a vocation, teaching management practices and the elements that contribute to good decision-making are just as important as instruction in the varieties and uses of mechanical aids to the farmer. The computer can be an important tool in the farmer's decision-making processes.

Despite the large part played by agriculture and related businesses in the economy of the United States, less than 10 percent of the development effort of computer manufacturers has been spent on products or systems related to agriculture. Fortunately, the principal development effort has occurred at colleges and universities, and the effort has produced many programs which are useful in the classroom.

The computer as a tool in farm record-keeping has many applications. It is used in analysis of farm enterprises to determine the profitability of each enterprise and whether the mixture of assets devoted to each enterprise is the optimum. In financial analysis, it can determine such things as whether to purchase or lease equipment. It is used in herd improvement programs. It determines such calculations as least-cost rations. Simulations of an entire farm can be made with the computer. Classroom adaptation of these programs can provide encapsulated experience which would require several years of actual production on the farm. The size of the farm and the factors affecting its operation can be changed at will in order to provide a variety of situations for the student's manipulation.

Computers are too expensive for individual farmers to own or lease. Most agriculturally related programs are written to serve a number of users and to be processed in some centrally located computer facility, which additionally benefits agriculture students. Many schools now have access to a computer network through terminals within the school, and many more can access a computer by batch processing (submitting input data on coding sheets which are keypunched into cards and read into the computer, then waiting for the output until the batch is processed). Many of the programs useful in the Vocational Agriculture classroom can be operated in either of the above modes.

It should be recognized that teaching a student to use service-oriented computer programs does not necessarily teach the concepts that can be used in later problem solving. The programs can be used, however, to reinforce theoretical and analytical concepts and provide experience in performing managerial functions.* Use of the computer also can eliminate a great deal of tedious data manipulation and calculation which contribute little to the education process but take up valuable classroom time. The added enhancement of the education process through "learning by doing" as the student "operates" his farm with a simulation program is still another advantage of the computer as an educational tool.

*Michael Boehlje, Vernon Eidman, and Odell Walker, "An Approach to Farm Management Education," *American Journal of Agricultural Economics*, 55 (May 1973).

USES OF THE COMPUTER IN VOCATIONAL AGRICULTURE

Drill and Practice

Drill and practice would contribute little to the Vocational Agriculture educational program. It is not the kind of curriculum in which the development of fundamental skills is the objective; the goal, rather, is to develop the ability to apply knowledge to the solution of problems and to establish principles and practices. Necessary drill and practice could be carried out as a part of other classroom activities and need not employ the unique capabilities of the computer.

Tutorial

Use of the computer as a tutorial aid has not become an important factor in Vocational Agriculture education. True, some areas in the curriculum are amenable to this technique, but little if any work has been done. Many other media adapt themselves to the tutorial or "self-learning" technique: programmed texts, film loops, audio-tape/slide sets, and similar devices. The proliferation and ready availability of these tools no doubt make them more popular than the computer for tutorial applications, and without doubt they are easier to acquire and use for the school which does not have ready access to computer equipment.

Problem Solving

One of the ways in which the computer can be used to good advantage in the Vocational Agriculture curriculum is as a problem solver. Many problems encountered in the management of complex enterprises like today's farms require the compilation and comparison of data involving many calculations or many factors. The computer's unique capability for accumulating and manipulating data makes it easy to solve this kind of problem.

Some useful computer programs are:

A program to calculate the future value of an investment at a fixed rate of compound interest.

A program to prepare a projected income statement for a planned enterprise. This program might also prepare a summary of costs and returns for an actual enterprise after it is completed.

A program which will calculate the pay-off period of an intended purchase of additional milk base for a given price per pound.

A program which will calculate the most profitable way of depreciating equipment and print the depreciation schedule.

A program which compares the costs of ownership and custom hire.

A program to calculate the true interest rate charged for money borrowed.

A program which calculates the amount for each equal payment needed to pay off a loan whose amount, term, and interest rate are known.

A program which will calculate the break-even point for rented land.

These are examples of the many kinds of problem solving programs available. Their value to the classroom teacher lies in eliminating complex and tedious calculations required to support the concept being taught: it permits the students' attention to be directed to the concept instead of being distracted by the mathematics involved.

It may be argued that students should learn to perform these calculations on their own, but it also can be argued that so much needs to be taught in so short a time that any increase in efficiency in the learning process is worth considering. When he comes to apply the concepts he has learned to real-life problems, the student will have access to numerous sources to solve them, and it may be more valuable for him to learn about these sources than to use valuable classroom time learning to work the problem himself. For example, a program which calculates the break-even analysis for rented land requires the following data: acres of land, length of crop rotation cycle, gross sales, hours of labor, tractor hours and cost per hour, fertilizer costs, side-dressing cost, seed cost, and all other costs. It is easy to imagine the opportunities for classroom discussion about the many problems and alternatives faced by the farmer in preparing the data for this program. The output is a tabular budget statement showing the per-acre costs for each expense item, the estimated value of the crops produced, and the maximum amount that can be paid per acre for the rented land. By simplifying the necessary arithmetic, the instructor is able to concentrate on the two important aspects of the problem: gathering the required data and analyzing the results.

Data Analysis

Data analysis programs in agriculture compare the results of problem-solving algorithms involving different conditions or strategies and select the best alternative. One such program might analyze the profitability of a proposed investment. Input includes the estimated life of the item, percent return the money can earn if invested elsewhere, income tax bracket of the operator, current interest rate for loans, depreciation method, type of loan, length of loan in years, total number of installments to repay the loan, purchase price of the item, down payment and/or trade-in, salvage value, additional income to be realized, reduced costs, and added costs. The output evaluates the profitability of the investment expressed as a change in the net worth of the business, the optimum depreciation schedule, the depreciation amount for each year of the life of the investment, and the break-even interest rate.

Data analysis has so many characteristics in common with simulations as applied in the agriculture curriculum that they tend to overlap. The data being analyzed will frequently represent an enterprise of the farm or a component of the enterprise. The program which analyzes the data thus becomes a simulation of the enterprise itself. For this reason, the remainder of the discussion deals with simulations.

Simulations

Agriculture is especially adaptable to simulations. It operates within a definite cycle—the growing season—of the same length every year for a given enterprise. The elements affecting the enterprise are relatively easy to isolate and quantify. The typical farm is made up of a number of different enterprises; each can be simulated separately, or the whole can be simulated. The typical livestock operation lends itself to simulation for the same reasons. Since the production cycle for the farm is so long, simulations have special value. The experience of several years' operation of a farm can be telescoped into weeks or days through computerized simulation. Games—simulations structured in such a way that they can be used competitively—are also useful.

Some samples may help the reader understand the scope and range of this kind of program.

Cash flow analysis is a simulation of the enterprise in financial terms. Input called for in inventory of assets, crop productions, sales purchases, livestock rations and feed use, income and expenses, capital acquisitions and sales, and personnel data. The output includes repayment schedule for current debts, cash flow report, profit-and-loss statement, net worth statement, inventory reports, and a highlights report showing (1) changes in net worth, (2) change in debt, (3) effect of 10 percent price increase, (4) effect of a 5 percent change in cash costs, (5) effect of a 10 percent drop in crop yield, (6) effect of a 10 percent drop in feed efficiency.

The cow game simulates selective breeding in livestock. Initially, 50 cows are bred to 5 bulls. Based on weaning weights, yearling weights, and conformation, the participants select replacements and determine the breeding program. The factors affecting the product (the calf) have been limited to those directly connected with breeding and selection in order to concentrate on a single aspect of the operation.

Several kinds of enterprise analysis programs are available. One, which has been adapted for dairy, beef ranch, or orchard enterprises, analyzes the data furnished by a group of operators about their enterprises. It compares each operator's enterprise with the group average. Report output includes financial summaries; labor, capital, and land management; and management of the animals, where appropriate.

A farm management game simulates a 640-acre irrigated crop farm with its appropriate machinery and equipment. Decisions are made regarding crop choice, land purchase, fertilizer, irrigation, and machinery. Price and yield are variables which are affected by the decisions and which must be considered in making the decisions.

Farm organization LP is a linear programming model used to determine the optimum combination of crops and crop acreages for individual farms. The input forms are self-explanatory and completely describe the individual farm.

Classroom use of the simulation, game, or data analysis program could be confined to dramatizing a concept or providing a jumping-off place for discussion; at the opposite extreme, operating a simulated farm could be a long-term project for an entire class, using the capabilities of one of the linear programming simulations.

Many simulation programs are still in use as batch programs. Some of the value of immediacy is lost by this method, but it does make the programs available to more users and—perhaps—helps to teach some of the patience necessary for those who are going to live with the long production cycle inherent in agriculture.

SUMMARY AND CONCLUSIONS

In summary, computer aids to education in the Vocational Agriculture curriculum are likely to include programs which provide opportunities to study farm operation in the classroom or which, by relieving the instructor of having to order and structure large amounts of data or perform complex calculations, simplify the development of understanding and familiarity with the decision-making process as applied to agriculture production. Computer programs need not provide instantaneous response to be usable as instructional aids, and batch processing, common in programs designed for agricultural users, will probably be with us for some time to come.

If current trends continue, computers will become increasingly available and usable in high schools. Modern educational methods are moving more and more in the direction of involving the student in the educational process. The computer provides a useful tool to encourage this involvement.

REFERENCES

Barton, Richard F., *A Primer on Simulation and Gaming.* Englewood Cliffs, N.J.: Prentice-Hall, 1970.

Boehlje, M., Eidman, V., and Walker, O., "An Approach to Farm Management Education." *American Journal of Agriculture Economics, 55* (May 1973), pp. 192–197.

Fuller, Earl L., "The Computer and Remote Terminals as Teaching Aids in Extension Management Education." Paper delivered at the North Central Farm Management Extension Workshop, Michigan State University, May 21–24, 1973.

Harsh, Stephen B., "The Application of Minimum-data Voice-response Programs to Management Problems." Paper delivered at the North Central Farm Management Extension Workshop, Michigan State University, May 21–24, 1973.

Hawkes, Nigel, *The Computer Revolution.* New York: E. P. Dutton, 1972.

Koch, Warren J., *The Use of Computers in Instruction in Secondary Schools.* Washington, D.C.: National Association of Secondary School Principals, 1972.

Margolin, Joseph B., and Misch, Marion R., ed., *Computers in the Classroom.* New York: Spartan Books, 1970.

User's Guide, Computerized Management Network. Blacksburg, Va.: Department of Agriculture Economics, Virginia Polytechnic Institute, 1972.

Zuckerman, David, and Horn, Robert E., *Guide to Simulation Games for Education and Training.* Lexington, Mass.: Information Resources, Inc., 1973.

APPENDIXES

APPENDIX I. Keys to Recognizing General Purpose Languages

KEYS TO RECOGNIZING GENERAL PURPOSE LANGUAGES

INTRODUCTION

It is not likely that you will be called upon to recognize programs written in all of the main languages mentioned in the first part of this booklet—FORTRAN, COBOL, ALGOL, PL/1, BASIC, APL and LOGO. It is particularly unlikely that you will want to use programs written in COBOL or any of the numerous other languages we have not mentioned here. To simplify your task of learning to recognize the language used for a program, we will take a closer look at the four languages you are most likely to encounter as a teacher—BASIC, FORTRAN, ALGOL, and PL/1.

Programming languages are structured very similarly, though they appear to be quite different. For example, the same three elements (input, process, and output) occur in each language, but the actual words used (e.g. READ or PRINT) may be different in each. The differences make it easier to recognize the language.

The next few pages contain examples of programs written in BASIC, FORTRAN, ALGOL, and PL/1. For each language, we point out the commonly used word for each type of instruction and some keys to help you recognize the language. It won't be necessary to memorize the keys: just read each one, compare it to the sample program in the language, and note the characteristics of the language.

BASIC

First, BASIC. This is the language used to write the programs for most of the computer-based curriculum units you will encounter as you look for useful programs for your own classes. You need not learn to write programs in BASIC or any other language; to use the units, you need only *recognize* the language and know whether it is available on your computer.

The distinguishing characteristics of BASIC are listed on page 189. Read over the list and then examine the BASIC program on page 189, noting the key features of BASIC.

FORTRAN

While BASIC is used to write programs for interactive mode, FORTRAN is commonly used in instructional materials where batch processing is appropriate.

Compare the FORTRAN program below with the same program written in BASIC. Notice that different words are used for the various types of instructions. Can you find the input, processing, and output instructions? How does the programmer make "remarks" in FORTRAN?

BASIC

```
10   REM  THIS IS AN EXAMPLE: A
20   REM  PROGRAM TO ADD 2 NUMBERS
25   PRINT "TYPE 2 NUMBERS"
30   INPUT X,Y
35   IF X=0 THEN 60
40   LET S=X+Y
50   PRINT "THE SUM IS"; S
55   GO TO 25
60   END
```

FORTRAN

```
C    THIS IS AN ADDITION PROGRAM
C    IT IS A FORTAN EXAMPLE
7    READ (2,12) FIRST, SEC
12   FORMAT (F10)
     SUM=FIRST + SEC
     WRITE (3,20) SUM
     GO TO 7
20   FORMAT (I11)
     END
```

Sample program in BASIC and FORTRAN.

KEYS TO BASIC

Input instructions. The word **INPUT** is commonly used in BASIC to request input from the user in interactive mode. Another input instruction is **READ**.

Processing instructions. The word **LET** often identifies processing instructions in BASIC. Some versons of BASIC, however, omit the word LET so that

```
60 LET R=I+2
```

would simply appear as

```
60 R=I+2
```

Output instructions. The word **PRINT** is primarily used for output in BASIC when the program is in interactive mode. Another output instruction is **WRITE**.

Programmer remarks. The word **REM** introduces a programmer comment in BASIC. These programmer remarks are for the use of anyone examining the program listing, and are ignored by the computer.

Line numbers. Every line in a BASIC program is numbered sequentially, often (but not always) by tens.

Loop instructions. Loop instructions cause the program to pass through the same set of instructions several times. The words **FOR** and **NEXT** are used in BASIC for loops. Study the two examples of loops below to see how loops work:

this loop produces

```
10 FOR I=1 TO 5
20 PRINT I
30 NEXT I
```

this output

```
1
2
3
4
5
```

this loop produces

```
60 FOR J=1 TO 5
70 PRINT "A"
80 NEXT J
```

this output

```
A
A
A
A
A
```

Loops can also be formed using a **GO TO** instruction.

Variable names. Look in processing instructions for names of numerical quantities. These names are restricted in BASIC to a single letter or a letter followed by one digit. The following names would be typical for quantities in BASIC programs: **A, B, X, Y, A1, Z0, V5**. Therefore, you would not find the form 10 SUM=X+Y, but rather **10 S=X+Y**.

Other keys. Look for the words **DIM** and **IF...THEN** in BASIC programs.

"DIM"

Programmer remarks

FOR...NEXT loop

PRINT - output instruction.

single letter variable name

```
10 DIM G(7,5),S(7),A(7)
15 REM  THIS PROGRAM COMPUTES AVERAGES
16 REM  FOR A SET OF STUDENT TEST SCORES.
20 FOR R=1 TO 7
30 FOR C=1 TO 5
40 READ G(R,C)
50 NEXT C
60 NEXT R
70 FOR R=1 TO 7
80 LET S(R)=0
90 LET A(R)=0
100 FOR C=1 TO 5
110 LET S(R)=S(R)+G(R,C)
120 NEXT C
130 LET A(R)=S(R)/5
140 NEXT R
150 PRINT "STUDENT NO.","AVERAGE"
160 FOR R=1 TO 7
170 PRINT R,A(R)
180 NEXT R
190 DATA 65,80,95,78,86
191 DATA 60,88,95,50,78
192 DATA 86,81,63,95,95
193 DATA 95,87,80,93,98
194 DATA 87,83,78,83,87
195 DATA 84,98,65,79,83
196 DATA 90,95,90,85,80
200 END
```

Every line numbered, sequentially.

BASIC program showing some key features.

Now look at the program again. Were your observations correct?

Programmer remark: C (for "comment")
Input instruction: READ
Processing instruction: no word, just an equation.
Output instruction: WRITE
Control instruction: GO TO

```
C    THIS IS AN ADDITION PROGRAM
C    IT IS A FORTRAN EXAMPLE
7    READ (2,12) FIRST, SEC
12   FORMAT (I10)
     SUM=FIRST + SEC
     WRITE (3,20) SUM
     GO TO 7
20   FORMAT (I11)
     END
```

FORTRAN program with comments.

Now look back over the two programs and see if you can identify the FORTRAN version of the input, process, and output instructions used in the BASIC program. Write the FORTRAN instructions in the space provided.

BASIC	*FORTRAN*
30 INPUT X, Y	______
40 LET S=X+Y	______
50 PRINT "THE SUM IS"; S	______

In the BASIC program, the programmer calls the two numbers *X* and *Y* and calls their sum *S*. What are they called in the FORTRAN version? Check your answers below.

Answers to questions above:

BASIC	*FORTRAN*
30 INPUT X,Y	7 READ (2,12) FIRST, SEC
40 LET S=X+Y	SUM=FIRST+SEC
50 PRINT "THE SUM IS"; S	WRITE (3,20) SUM

"X" in BASIC is "FIRST" in FORTRAN; "Y" in BASIC is "SEC" in FORTRAN; "S" in BASIC is "SUM" in FORTRAN.

At this point, you can probably recognize certain identifying features of FORTRAN. Here is a summary:

KEYS TO FORTRAN

Input instructions. The word **READ** is primarily used for input in FORTRAN, although other words are also used.

Processing instructions. No special word like LET is used for processing. These statements usually take the form of an equation like:

```
A(I)=A(I+8)
NET=GROSS-DED
TEMP=K
```

Output instructions. The word **WRITE** is usually used for output in FORTRAN, although other words such as **PRINT** are also used.

Programmer remarks. **C**, short for "Comment", allows the programmer to make remarks in the program which will be ignored by the computer.

Line numbers. Not every line need be numbered in a FORTRAN program. Line numbers in FORTRAN also do not have to appear in sequence.

Loop instructions. Loops are made in FORTRAN programs with the words **DO** and **CONTINUE**. For example, the following loop would produce the indicated output:

this loop produces

```
   DO 10 I=1,5
   WRITE (3,15) I
15 FORMAT (1H ,I2)
10 CONTINUE
```

this output

```
1
2
3
4
5
```

Loops can also be formed using a **GO TO** instruction.

Variable Names. Most versions of FORTRAN allow variables to be named with up to six characters. The following names would be acceptable in FORTRAN: **ACCNT, SUM, SUM3A, A8O4, RESULT.**

Other keys. Look for the words **FORMAT**, and **DIMENSION** in FORTRAN programs.

Check to see that you understand the key features of FORTRAN by looking at the sample program below.

Programmer remark

Recognizable variable name (6 letters)

```
C     INITIAL VALUES FOR VARIABLES
      OLOAN=32700.00
      RATEM=0.075/12.0
      PAYMNT=220.00
      MONTH=3
      IYR=70
      TOTINT=0.00
      DO 4J=1,360
C     STEP 1.  COMPUTE INTEREST PAYMENT
      PDINT=OLOAN*RATEM
C     STEP 2.  COMPUTE PAYMENT OF PRINCIPAL
      PRINPT=PAYMNT-PDINT
      WRITE (3,2) OLOAN,PRINPT,PDINT,MONTH,IYR
    2 FORMAT (' ',F8.2,F8.2,F8.2,4X,I2,'/',I2)
C     ACCUMULATE INTEREST PAID SO FAR
      TOTINT=TOTINT+PDINT
      IF (OLOAN-PRINPT) 1,1,6
C     STEP 3.  REDUCE PRINCIPAL
    6 OLOAN=OLOAN-PRINPT
      MONTH=MONTH+1
      IF (12-MONTH) 5,4,4
    5 MONTH=1
      IYR=IYR+1
    4 CONTINUE
C     PRINT LINE AT END OF REPORT
    1 WRITE (3,3) OLOAN,TOTINT
    3 FORMAT ('1BALANCE OF LOAN=',F8.2,F10.2)
      STOP
      END
```

Loop

WRITE = output instruction

FORMAT statement

Not all lines numbered; not numbered sequentially.

FORTRAN program showing key figures.

Reading each line, you can easily spot the programmer remark explaining what the program is for. Can you also find an input instruction, a processing instruction, an output instruction, and a control instruction? Can you identify a loop?

Try to find each of these features. When you have identified as many of the program elements as possible, see below, where the distinguishing characteristics of ALGOL are described.

ALGOL

Let's see now what our simple addition program would look like if it were written in ALGOL:

```
BEGIN
COMMENT     ADDITION PROGRAM;
INTEGER     NUM1, NUM2, TOTAL;
LABEL       HERE;
FILE        CARD (KIND=READER, MAXRECSIZE=14);
FILE        LINE (KIND=PRINTER, MAXRECSIZE=22);
FORMAT      FM1 (I10);
FORMAT      FRMT (I11);
NOW:        READ (CARD, FMT, NUM1, NUM2);
            TOTAL:=NUM1 + NUM2;
            WRITE (LINE, FRMT, TOTAL);
            GO TO NOW;
END.
```

Sample program in ALGOL.

KEYS TO ALGOL

Input instructions. A commonly used input instruction in ALGOL is **READ**; however, this word is not standard for all versions of ALGOL.

Processing instructions. Probably the best way to instantly recognize an ALGOL program is to find a processing instruction. ALGOL is the only commonly used language that does not use a simple equals sign (=) in the processing instruction. Instead, a special symbol is used--a colon followed by an equals sign, :=. For example,

TOTAL :=NUM1 + NUM2

In another version of ALGOL, a "back arrow" ←, is used in place of the (:=), like this:

TOTAL←NUM1 + NUM2

Output instructions. Again, there is no standard output instruction in ALGOL. **WRITE** is commonly used, but is not found in all versions of ALGOL.

Programmer remarks. The word **COMMENT** allows the programmer to insert remarks which will be ignored by the computer.

Line numbers. Occasionally, a line will be "labelled" with a word followed by a colon (like **NOW:**) but not with a line number.

Loop instructions. The following combinations of words create loops in ALGOL:

FOR...STEP...UNTIL...DO...

WHILE...DO...

DO...UNTIL...

Loops can also be formed using a **GO TO** instruction.

Variable Names. Names in ALGOL usually can be much longer than in FORTRAN or BASIC; names up to 15 characters long (such as **THELASTRESULT**) are often allowed.

Other keys. Many of the lines in an ALGOL program may end with a semicolon ;. Some versions of ALGOL allow lower-case letters to be used (e.g., a variable could be named, "thelastresult"). ALGOL programs are divided into a series of blocks or segments, characterized by the word **BEGIN** at the start and **END** at the end. A long program may have dozens of **BEGIN** and **END** instructions.

Check the features of ALGOL in the program below.

"BEGIN" instruction
Programmer remark
; at end of line
Line label
:= processing sign
"END" instruction
Long variable name

```
           BEGIN
COMMENT         OXYGEN (PERCENT OF OXYGEN IN POTASSIUM CHLORATE) PROGRAM;
REAL              MASSE,MASSB,MASSA,MASSU,MASSCL2,MASSO,EPCT,TPCT,ERROR,
                  PCTERROR;
INTEGER           ID;
ALPHA             SN1,SN2,TN1,TN2;
LABEL             START,DONE;
FILE              CARD(KIND=READER,MAXRECSIZE=14),

                  LINE(KIND=PRINTER,MAXRECSIZE=22);

FORMAT            FI1(4(A5) ,X5,I5,3(X4,F6,2));
FORMAT            FO1(X32,"MASS OF CR+COVER  CR+COVER   MASS OF   MASS OF"
                  "  MASS OF   EXPER   THEOR");
FORMAT            FO2(X24,"RUN   EMPTY CR.  +KCLO3    +KCLO3     KCLO3  "
                  "   KCLO2    OXYGEN     % OF     % OF           "
                  "PERCENTAGE");
FORMAT            FO0(X45,"PERCENT OF OXYGEN IN POTASSIUM CHLORATE"//);
FORMAT            FO3("TEACHER   STUDENT      ID      + COVER  BEFORE HT "
                  "AFTER HT    USED    PRODUCED  DRVN OFF   OXYGEN    "
                  "OXYGEN     ERROR     ERROR");
FORMAT OUT        FO4("--------------------------------------------------"
                      "--------------------------------------------------"
                      "--------------------------------",/);
FORMAT OUT        FO5(2A5,X1,2A5,X2,I5,10,X4,F6,2)/);
                  WRITE(LINE,FO0);WRITE(LINE,FO1);
                  WRITE(LINE,FO2);
                  WRITE(LINE,FO3);
                  WRITE(LINE,FO4);
START:         READ(CARD,FI1,SN1,SN2,TN1,TN2,ID,MASSE,MASSB,MASSA)[DONE];
              MASSU := MASSB - MASSE;
                  MASSCL2 := MASSA - MASSE;
                  MASSO := MASSB - MASSA;
                  EPCT := MASSO/MASSU*100;
                  TPCT := 39.17;
                  ERROR := ABS(TPCT - EPCT);
                  PCTERROR := ERROR/TPCT*100;
               WRITE(LINE,FO5,TN1,TN2,SN1,SN2,ID,MASSE,MASSB,MASSA,MASSU,
                  MASSCL2,MASSO,EPCT,TPCT,ERROR,PCTERROR);
                  GO TO START;
DONE:          END
```

ALGOL program showing key features.

PL/1

Finally, look at the addition program written in the PL/1 language:

```
      /*ADDITION AGAIN, FOR THE LAST TIME*/

START:   GET LIST (N1,N2);
         TOTAL = N1 + N2;
         PUT LIST (TOTAL);
         GO TO START;
         END;
```

Addition program in PL/1.

Note the programmer's remarks set off by slashes and asterisks(/*).

KEYS TO PL/1

Input instructions. The word **GET** is frequently used in PL/1 programs for input instructions. **READ** is also used.

Processing instructions. As in FORTRAN, processing instructions are usually in the form of an equation, like

ALL=PART1 + PART2

Like other languages, PL/1 can assign the same value to several different variables, but these "multiple assignments" are done in a unique way. Compare the instructions below, noting how PL/1 differs from the others.

```
BASIC      X=Y=0      }  Each of these instructions results
FORTRAN    X=Y=0      }  in assigning the value 0 to both
ALGOL      X:=Y:=0    }  the quantities X and Y. (Note the
PL/1       X,Y=0      }  comma in PL/1.)
```

Output instructions. **PUT** is used in many PL/1 programs for output; other words are also used, such as **WRITE**.

Programmer remarks. These are inserted between slashes and asterisks:

/*THIS IS A TYPICAL PL/1 COMMENT*/

Line numbers. As in ALGOL, lines of PL/1 programs are labelled with a word followed by a colon; for example, **A700:** and **HERE:** are typical labels.

Loop instructions. The following words are used to form loops in PL/1:

DO...END
DO...TO...BY...END
DO...WHILE...END

Variable names. Most versions of PL/1 allow names up to 31 characters.

Other keys. Look for the words **DECLARE** and **DCL** in PL/1 programs. Also look for many punctuation marks (such as semicolons) at the ends of PL/1 instructions.

Look at the segment of a PL/1 program below. How quickly could you identify it using the key to PL/1 listed on the previous page?

```
Programmer      STATISTICAL SUMMARY: /* THIS PROGRAM OBTAINS AND PRINTS A MINIMUM,
remark, /*         MAXIMUM, MEAN, AND STANDARD DEVIATION FOR EACH DATA SAMPLE*/
                      PROCEDURE OPTIONS (MAIN);
"DECLARE"                  DECLARE 1 DATUM, 2(CODE CHAR (2), X FIXED (4,2)), SUMX
                              FIXED (8,2), SUM XSQ FIXED (12,4), (XMAX, XMIN) FIXED
                              (4,2), OLD CODE CHAR (3), N FIXED (4), DATA FILE,
                              SWITCH LABEL INITIAL (NEXT SAMPLE);
                           ON ENDFILE (DATA) BEGIN;
                                                 SWITCH = END UP;
                                                 GO TO CALC WRITE;
"GET" -- input                                   END;
instruction                GET FILE (DATA) LIST (DATUM);
                   NEXT SAMPLE:  /*INITIALIZE AND EXECUTE THE DO GROUP BELOW FOR
                      EACH NEW SAMPLE */
multiple value             OLD CODE = CODE;  /* CODE=SAMPLE-CODE PREFIX */
assignment                 XMAX, XMIN = X;
                           SUMX, SUM XSQ, N =0;
                           DO WHILE (CODE = OLD CODE); /*DO GROUP WILL ITERATE UNTIL*/
                              N = N+1;                 /*A NEW SAMPLE IS ENCOUNTERED*/
                              SUMX = SUMX+X;
; at ends of                  SUM XSQ = SUM XSQ+X**2;
lines                         IF X < XMIN
                                 THEN XMIN = X;
                              ELSE IF X > XMAX
                                 THEN XMAX = X;
                              GET FILE (DATA) LIST (DATUM);
                           END; /* END OF DO GROUP */
                   /* SUBRTN. TO OBTAIN MEAN & STAND. DEV., THEN PRINT*/
                           DECLARE MEAN FIXED (4,2), ST DEV FIXED (5,2);
line label         CALC WRITE: MEAN = SUMX/N;
                           ST DEV = SQRT ((SUM XSQ-N*MEAN**2)/(N-1));
                           PUT LIST ('CODE:  ', OLD CODE, '   SAMPLE SIZE: ',N,
                              '  MAX: ',XMAX,'  MIN: ', XMIN,'  ST.DEVIATION: ',
"PUT" output                  ST DEV, '  MEAN: ', MEAN) SKIP (2);
instruction                GO TO SWITCH;
                   END UP: PUT LIST ('END OF DATA') SKIP (5);
                      END;  /*END OF PROGRAM */
```

PL/1 program showing key features.

A FLOW CHART FOR LANGUAGE IDENTIFICATION

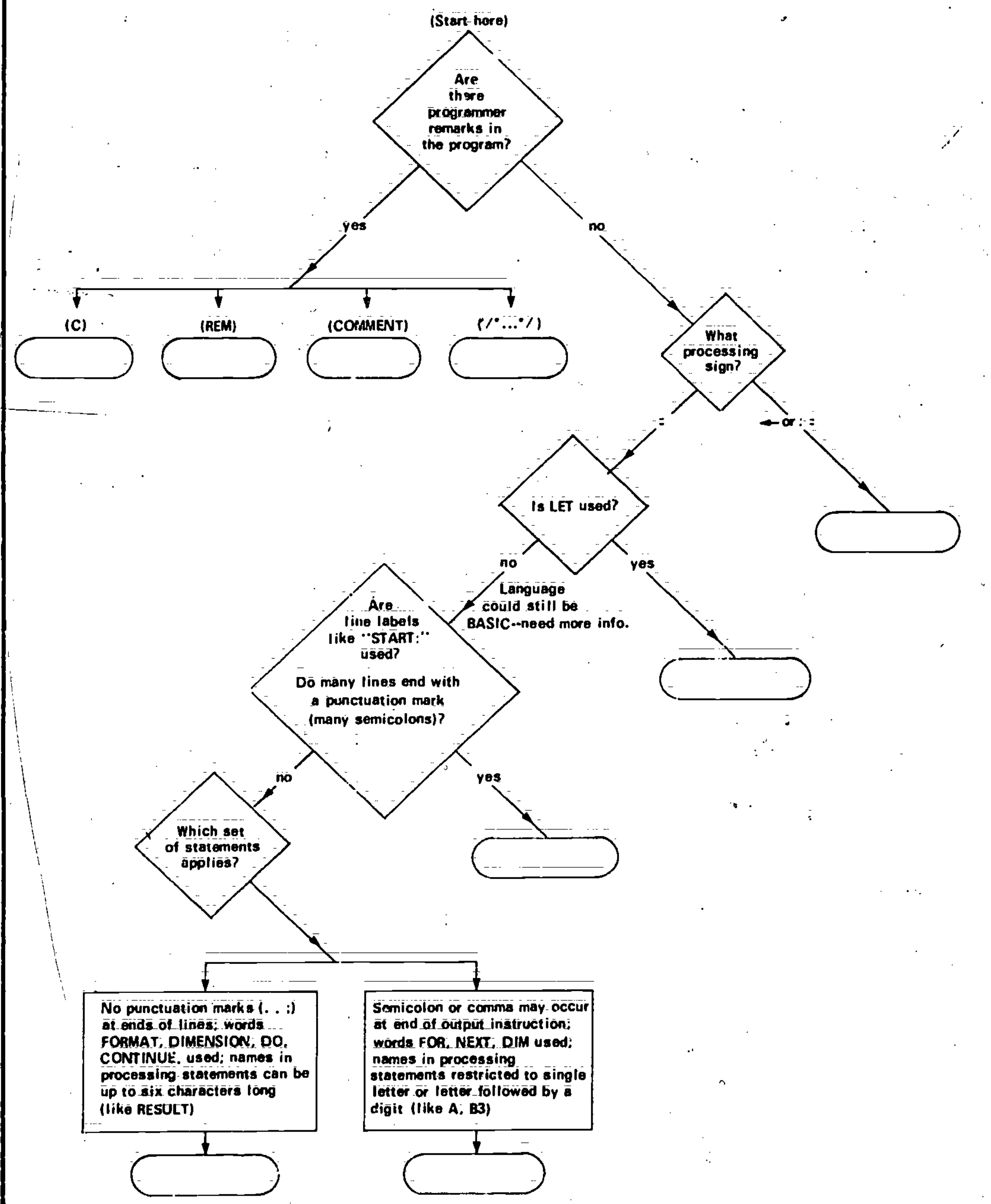
At this point you may be somewhat confused, especially if this is your first experience with computer programming languages. You might be asking yourself questions like, how can I tell the difference between ALGOL and BASIC if they both use READ statements? Or, what if all the lines in a program are numbered? The program can be BASIC, but it also can be FORTRAN!

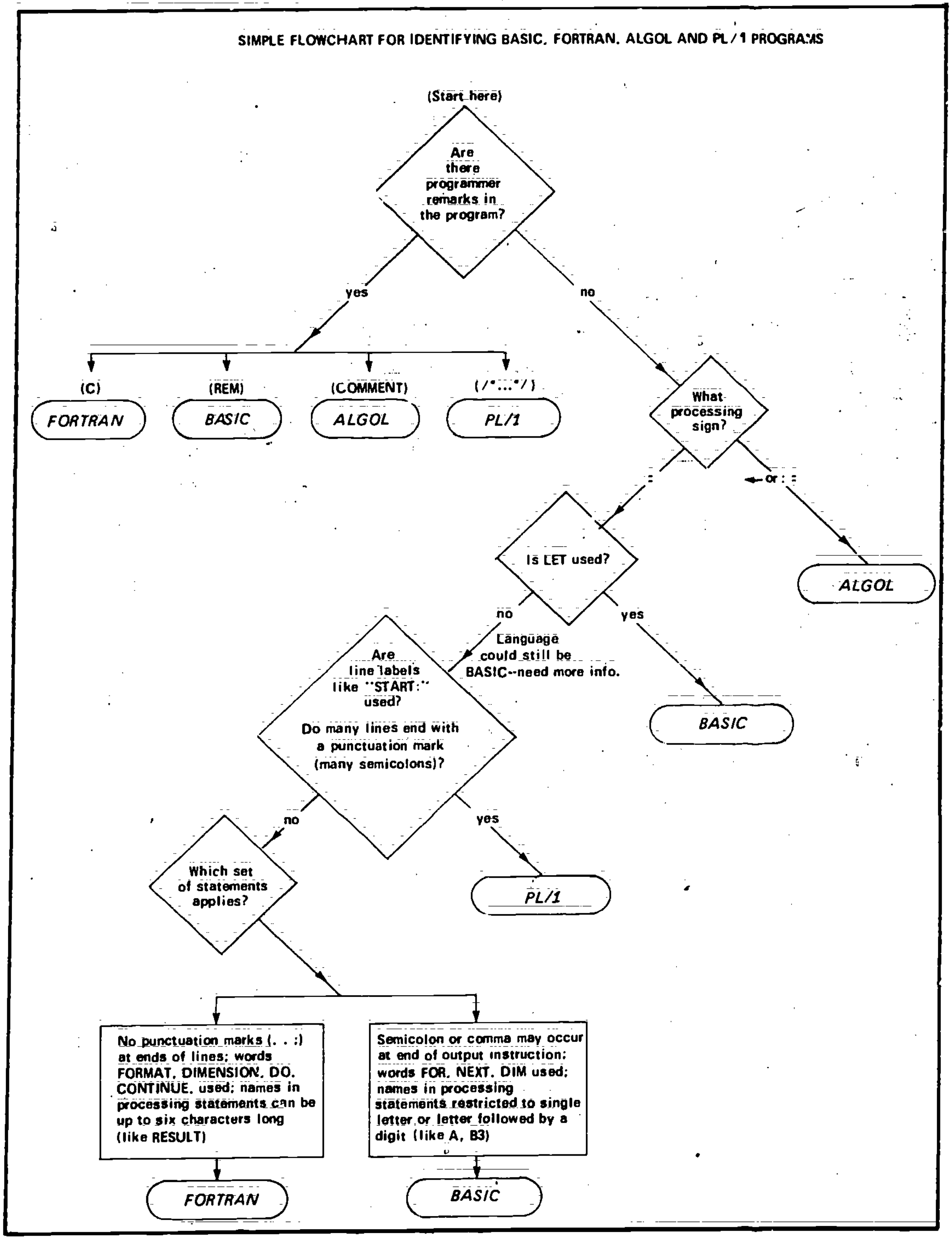
To help summarize all the information you have read so far, try to fill out the flow chart on page 194. The chart is a simple system for identifying the four programming languages we have discussed. The ovals (⬭) are meant to contain the name of one of the four languages: BASIC, FORTRAN, ALGOL, or PL/1. Refer back to the keys for each language if you want to. Then check your answers with those shown on page 195.

PRACTICE RECOGNIZING LANGUAGES

Now practice your skill at recognizing languages by correctly labelling each of the following eight programs. Use the flow chart on page 195 to help you.

SIMPLE FLOWCHART FOR IDENTIFYING BASIC, FORTRAN, ALGOL AND PL/1 PROGRAMS




SIMPLE FLOWCHART FOR IDENTIFYING BASIC, FORTRAN, ALGOL AND PL/1 PROGRAMS
(Start here)
Are there programmer remarks in the program?
yes
no
(C)
FORTRAN
(REM)
BASIC
(COMMENT)
ALGOL
(/*...*/)
PL/1
What processing sign?
=
or :=
ALGOL
Is LET used?
no
yes
Language could still be BASIC--need more info.
BASIC
Are line labels like "START:" used?
Do many lines end with a punctuation mark (many semicolons)?
no
yes
PL/1
Which set of statements applies?
No punctuation marks (. . ;) at ends of lines; words FORMAT, DIMENSION, DO, CONTINUE, used; names in processing statements can be up to six characters long (like RESULT)
Semicolon or comma may occur at end of output instruction; words FOR, NEXT, DIM used; names in processing statements restricted to single letter or letter followed by a digit (like A, B3)
FORTRAN
BASIC

As is evident from the flow chart, there are actually very few items within the program which you should have to check before determining which language is being used.

> *A Word of Caution:* When you identify languages of programs outside of this book, it is possible that the flow chart will not enable you to identify a language immediately. Use it to help you conclude that a language "looks like FORTRAN," or "appears to be ALGOL," but go back and check the keys before making any definite conclusions. There are many programming languages, and usually you can find common elements between any two. If a language *looks like* FORTRAN (but really isn't), chances are it can be translated into FORTRAN by a programmer without too much trouble. Keep in mind also that there are many different versions of the same language; for example, the BASIC used on a Hewlett-Packard computer is not exactly the same as the BASIC used on Control Data Corporation computers. So, even if you are positive that a certain program is written in BASIC, let an experienced programmer look at it to determine what changes (if any) must be made so the program will run on your computer system.

(1) ______________
(Language)

```
            BEGIN
COMMENT        OXYGEN (PERCENT OF OXYGEN IN POTASSIUM CHLORATE) PROGRAM;
REAL              MASSE,MASSB,MASSA,MASSU,MASSCL2,MASSO,EPCT,TPCT,ERROR,
                  PCTERROR;
INTEGER           ID;
ALPHA             SN1,SN2,TN1,TN2;
LABEL             START,DONE;
FILE              CARD(KIND=READER,MAXRECSIZE=14),

                  LINE(KIND=PRINTER,MAXRECSIZE=22);

FORMAT            FI1(4(A5),X5,I5,3(X4,F6.2));
FORMAT            FO1(X32,"MASS OF CR+COVER  CR+COVER   MASS OF   MASS OF"
                  "   MASS OF    EXPER    THEOR");
FORMAT            FO2(X24,"RUN   EMPTY CR.  +KCLO3    +KCLO3     KCLO3  "
                  "   KCLO2    OXYGEN     % OF      % OF             "
                  "PERCENTAGE");
FORMAT            FO0(X45,"PERCENT OF OXYGEN IN POTASSIUM CHLORATE"//);
FORMAT            FO3("TEACHER   STUDENT      ID    + COVER BEFORE HT "
                  "AFTER HT   USED    PRODUCED  DRVN OFF   OXYGEN      "
                  "OXYGEN    ERROR     ERROR");
FORMAT OUT        FO4("--------------------------------------------------"
                      "------------------------------------------------"
                      "-------------------------------"/);
FORMAT OUT        FO5(2A5,X1,2A5,X2,I5,10(X4,F6.2)/);
                  WRITE(LINE,FO0);WRITE(LINE,FO1);
                  WRITE(LINE,FO2);
                  WRITE(LINE,FO3);
                  WRITE(LINE,FO4);
START:          READ(CARD,FI1,SN1,SN2,TN1,TN2,ID,MASSE,MASSB,MASSA)[DONE];
               MASSU := MASSB - MASSE;
                  MASSCL2 := MASSA - MASSE;
                  MASSO := MASSB - MASSA;
                  EPCT := MASSO/MASSU*100;
                  TPCT := 39.17;
                  ERROR := ABS(TPCT - EPCT);
                  PCTERROR := ERROR/TPCT*100;
                WRITE(LINE,FO5,TN1,TN2,SN1,SN2,ID,MASSE,MASSB,MASSA,MASSU,
                  MASSCL2,MASSO,EPCT,TPCT,ERROR,PCTERROR);
                  GO TO START;
DONE:            END.
```

(2) ____________________
(Language)

```
60   PRINT  'THIS PROGRAM SHOWS HOW A PIECE OF CAPITAL EQUIPMENT'
70   PRINT  'DEPRECIATES ACCORDING TO THREE COMMONLY USED DEPRECIATION'
80   PRINT  'METHODS; STRAIGHT LINE, SUM OF THE DIGITS, AND DOUBLE'
90   PRINT  'DECLINING.'
120  PRINT 'ORIGINAL COST';
130  INPUT C
140  PRINT 'LIFE OF ITEM';
150  INPUT L
160  PRINT ' SCRAP VALUE';
170  INPUT S
180  PRINT
190  PRINT 'YEAR','STRAIGHT','SUM OF','DOUBLE'
200  PRINT '   ',' LINE','DIGITS','DECLINING'
210  PRINT
220  LET V=C-S
230  LET D1=V/L
240  LET Y=((L+1)/2)*L
250  LET Z=L
260  FOR X=1 TO L
270  LET D2=V*(Z/Y)
280  LET Z=Z-1
290  LET D3=2*C/L
300  LET C=C-D3
310  PRINT X'
312  LET Q=D1
314  GOSUB 400
316  LET Q=D2
318  GOSUB 400
320  LET Q=D3
322  GOSUB 400
325  PRINT
330  NEXT X
350  STOP
400  LET Q=INT (Q*100)/100
420  IF Q>100 THEN 440
430  PRINT ' ';
440  IF Q>10 THEN 460
450  PRINT ' ';
460  PRINT '$'Q,
490  RETURN
999  END
```

(3) ______________

(Language)

```
C         PAYROLL PROGRAM
C         READ EMPLOYEES ID, HOURS WORKED AND PAY RATES
   10     READ (5,20) EMPID, HRS, RATE, ORATE
   20     FORMAT (2F5.0,2F5.2)
          IF (EMPID .EQ. 77777.0) GO TO 60
          IF (HRS .GT. 40.0) GO TO 50
C         COMPUTE WEEKLY PAY
          PAY = RATE * HRS
C         CONVERT EMPLOYEES ID TO FIXED POINT FOR PRINTOUT
   30     IEMPID = EMPID
C         PRINT EMPLOYEES ID AND WAGES
          WRITE (6,40) IEMPID, PAY
   40     FORMAT (12H1EMPLOYEE ID,I5,7H  PAY, F8.2)
          GO TO 10
C         COMPUTE OVERTIME PAY
   50     A = ORATE * (HRS - 40.0)
C         COMPUTE BASE PAY
          B = RATE * 40.0
C         COMPUTE TOTAL WAGES
          PAY = A + B
          GO TO 30
   60     STOP
          END
```

(4) ______________

(Language)

```
  /*THIS IS THE FIRST SEGMENT OF THE 'ONIONS' PROGRAM FROM NO. CAROLINA*/
  /*EDUC. COMPUTING SERVICE*/
START:  GET LIST(SACKS);
  IF SACKS <=Ø THEN GO TO FINISH;
  IF SACKS>5ØØØØ THEN PUT LIST
     ('YOU DO NOT HAVE THE RESOURCES TO PRODUCE',SACKS,'SACKS');
  IF SACKS>5ØØØØ THEN PUT LIST
     ('OF ONIONS.  TRY PRODUCING A SMALLER QUANTITY.');
  IF SACKS>5ØØØØ THEN GO TO START;
  LOG1=1Ø**.681*DISPIN**1.31;
  LOG3=1Ø**(.134+.Ø123*(TY-1924))*COST**-.512*WAGE**-1.2;
  UNHARV=1Ø**.13*WEALTH**2.4*(SACKS*.ØØ1)**2.S6;
  QUAND=SACKS-UNHARV;
  PRICE=LOG1*(QUAND*.ØØ1)**-2.27*.Ø1;
  QUANM=EXP (LOG(1/LOG3/LOG1**.324)/-1.7355);
  LOG2=-1.27*LOG1*QUANM**.33/-1.6;
  QUANM=QUANM*1ØØØ;
  COSTP=LOG2*(QUAND*.ØØ1)**-2.6*.Ø1;
  PROF=QUAND*(PRICE-COSTP);
  TOREV=QUAND*PRICE;
  IF PROF>=0 THEN GO TO OUTPUT;
  PUT LIST ('UNFORTUNATELY, EVE       UGH YOU SOLD',QUAND, 'SACKS OF');
  PUT LIST ('ONIONS, YOUR FIXED C     S WERE HIGH ENOUGH TO CAUSE YOU');
  PUT LIST ('TO SUFFER A LOSS.  T     AGAIN, BUT TRY TO PRODUCE MORE.');
  GO TO START;
```

(5) ________________
(Language)

```
                                    G A S 1
                                    = = = =

          BEGIN

COMMENT        GAS1  (VOLUME OCCUPIED BY A MOLE OF GAS) PROGRAM;

REAL           W,V,MOLESMG,CALCMV,MOLES,ERROR;
INTEGER        ID;
ALPHA          SN1,SN2,TN1,TN2;
LABEL          START,DONE;
FILE           CARD (KIND=READER,MAXRECSIZE=14);

FILE           LINE (KIND=PRINTER,MAXRECSIZE=22);

FORMAT         FI1 (4(A5),X5,I5,2(F10.5));
FORMAT         FO1 (X60,"GAS 1"/);
FORMAT         FO2 (X46,"VOLUME OCCUPIED BY A MOLE OF GAS"/);
FORMAT         FO3 (X33,"RUN",X6,"WEIGHT",X9,"VOLUME",X9,"MOLES",X10,
               "CALC.",X25,"PERCENT");
FORMAT         FO4 (X7,"TEACHER",X8,"STUDENT",X4,"ID",X9,"MG",X12,

               "GAS",X13,"MG",13,"MV",X12,"MOLS",X11,"ERROR");
FORMAT         FO5 (X5,"-----------------------------------------"
                       "-----------------------------------------"
                       "-----------------------------------------");
FORMAT         FO6 (2(X5,2A5),X2,I5,X3,4(F10.5,X5),E12.5,X5,F6.2/);
               WRITE (LINE, FO1);
               WRITE (LINE, FO2);
               WRITE (LINE, FO3);
               WRITE (LINE, FO4);
               WRITE (LINE, FO5);
START:         READ (CARD,FI1,SN1,SN2,TN1,TN2,ID,W,V)   [DONE];

               MOLESMG   := W/24.3;
               CALCMV    := V/MOLESMG;
               MOLES     := MOLESMG*6.02@ +23;
               ERROR     := (22.4-CALCMV)*100/22.4;
               WRITE (LINE,FO6,TN1,TN2,SN1,SN2,ID,W,V,MOLESMG,CALCMV,
               MOLES, ERROR);
               GO TO START;

DONE:     END.
```

(6) ______________
(Language)

```
C THE SUM OF TWO NUMBERS
      READ (2,3) I, J
    3 FORMAT (2I11)
C LIST THE INPUT
      WRITE (3,20) I,J
   20 FORMAT (1X, 6HINPUT , 2I11)
C FIND AND OUTPUT THE SUM
      K=I+J
      WRITE (3,16) K
   16 FORMAT (1X,I11)
      STOP
      END
```

(7) ______________
(Language)

```
10  REM   OVERTONE SERIES GENERATOR
20  PRINT "HOW MANY OVERTONES DO YOU WANT";
30  INPUT T
40  PRINT "WHAT IS THE FUNDAMENTAL FREQUENCY";
50  INPUT F
60  PRINT
70  PRINT "OVERTONE", "FREQUENCY", "RATIO"
80  FOR I=1 TO T
90  LET H=F*I
100 LET R=I
110 IF R<=2 THEN 140
120 LET R =R/2
130 GO TO 110
140 PRINT I,H,R
150 NEXT I
160 PRINT
170 END
```

(8) ____________________
(Language)

```
C71A3:   PROCEDURE OPTIONS (MAIN);
         DECLARE 1 CARDIN,
                   2 CODE CHARACTER (1),
                   2 ITEM CHARACTER (8),
                   2 QTY PICTURE 'ZZZ9'
                   2 REST CHARACTER (67);
      /* DECLARATION FOR STOCK FILE AND STOCK RECORD  */
          %INCLUDE F71SF,F71SR;
          DECLARE LOSS FIXED DECIMAL (10,2);
          DECLARE PAGE NO FIXED DECIMAL INITIAL (0);
      /* HEADING AND PAGE CONTROL */
          OPEN FILE (EXCP) PRINT PAGESIZE (50);
          ON ENDPAGE (EXCP) BEGIN;
               PAGE NO = PAGE NO + 1;
               PUT FILE (EXCP) PAGE LINE (3)
                EDIT ('PAGE',PAGE NO,'EXCEPTIONAL ADJUSTMENTS')
                     (X(10),A(4),F(5),X(20),A(25));
               PUT FILE (EXCP) LINE (6)
                EDIT ('ITEM NO','DESCRIPTION','VALUE','TYPE')
                     (COL(10),A(10),COL(25),A(12),COL(54),2 A(10));
               END;
      /* PRINT HEADINGS FOR FIRST PAGE */
         SIGNAL ENDPAGE (EXCP);
      /* SET UP ERROR CONTROL */
         ON ERROR BEGIN;
      /* TEST FOR KEY OR CONVERSION ERROR OR SIGNALLED ERROR*/
         IF ONCODE>49 &ONCODE<58|ONCODE>599&ONCODE<630|ONCODE=9
         THEN DO;
              WRITE FILE (ERRORS) FROM (CARDIN);
              GO TO READCARD;
              END;
      /* IF NOT KEY OR CONVERSION ERROR THEN DISPLAY INPUT AND END JOB*/
         DISPLAY (CODE||ITEM||QTY||ONCODE);
         END;
      /* SET UP NORMAL TERMINATION CONTROL*/
         ON ENDFILE (INFILE) GO TO FINIS;
      /* MAIN UPDATING LOOP  */
READCARD:  READ INTO (CARDIN) FILE (INFILE);
           READ FILE (STOCK) INTO (STREC) KEY (ITEM);
           IF CODE = 'A' THEN
      /*   ADJUSTMENTS TO STOCK LEVEL  */
            DO;
                  LOSS = (SQTY-QTY) * SPRICE;
                  IF LOSS>1000 THEN CALL XCEPT (LOSS,'S');
                  SQTY = QTY;
                  REWRITE FILE (STOCK) FROM (STREC) KEY (ITEM);
                  GO TO READCARD;
            END;
            IF CODE ¬='D'  THEN SIGNAL ERROR;  /*  ILLEGAL CODE */
     /*  DELETIONS */
           LOSS = SQTY * SPRICE;
           IF LOSS> 1000 THEN CALL XCEPT (LOSS,'O');
           DELETE FILE (STOCK) KEY (ITEM);
           GO TO READCARD;
XCEPT:     PROCEDURE (L,TYPE);
           DECLARE L FIXED DECIMAL  (10,2),
                   TYPE CHARACTER (1);
           PUT FILE (EXCP) SKIP
               EDIT  (ITEM,SDESC,L,TYPE)
                     (COL(10),A(8),COL(25),A(25),F(12,2),X(3),A(1));
           END XCEPT;
      /*   NORMAL TERMINATION  */
FINIS:   PUT FILE (EXCP) SKIP (2) LIST('END OF JOB');
END C71A3;
```

Answers

(1) ALGOL
(2) BASIC
(3) FORTRAN
(4) PL/1
(5) ALGOL
(6) FORTRAN
(7) BASIC
(8) PL/1

APPENDIX II. Some Primary Sources of Computer-Based Instructional Units

SOME PRIMARY SOURCES OF COMPUTER-BASED INSTRUCTIONAL UNITS

1. Dartmouth Secondary School Project
   Kiewit Computation Center
   Dartmouth College
   Hanover, New Hampshire 03755
2. Computing and Mathematics Curriculum Project
   University of Denver
   Denver Park, Colorado 80210
3. Computer Curriculum Project
   Hewlett-Packard Company
   333 Logue Avenue
   Mountain View, California 94040
   (Includes Project Solo material)
4. Dymax
   People's Computer Company
   P.O. Box 310
   Menlo Park, California 94025
5. Computer Technology Program
   Northwest Regional Educational Laboratory
   710 S. W. Second Avenue
   Portland, Oregon 97204
6. Digital Equipment Corporation
   146 Main Street
   Maynard, Massachusetts 01754
   (Includes Huntington I and II simulations)
7. Project Solo
   University of Pittsburgh
   Pittsburgh, Pennsylvania 15213
8. Educational Technology Publications
   140 Sylvan Ave.
   Englewood Cliffs, New Jersey 07632
9. TIES
   1925 West County Road B2
   Roseville, Minnesota 55113
10. MECC
    2520 Broadway Drive
    Lauderdale, Minnesota 55113
11. Data Processing Curriculum Group
    Science Research Associates, Inc.
    259 East Erie Street
    Chicago, Illinois 60611
12. North Carolina Educational Computing Service
    Research Triangle Park, North Carolina 27709
13. Tecnica/REACT
    Tecnica Education Corporation
    1864 South State Street
    Salt Lake City, Utah 84115
14. Data General Corporation
    Southboro, Massachusetts 01772
15. Education Center
    Honeywell Information Systems, Inc.
    110 Cedar Street
    Wellesley Hills, Massachusetts 02181
16. Computer Curriculum Corporation
    1032 Elwell Court
    Suite 106
    Palo Alto, California 94303
17. Coast Community College District
    Office of Educational Development
    1370 Adams Avenue
    Costa Mesa, California 92626
18. Florida State University
    Computer-Assisted Instruction Center
    Tully Building
    Tallahassee, Florida 32306

In addition, see:

Index to Computer Based Learning, 1976 Edition
Anastasia C. Wang, Editor
The Instructional Media Laboratory
The University of Wisconsin-Milwaukee
P.O. Box 413
Milwaukee, Wisconsin 53201

INDEX

A

B

C

D

E

M

N

O

P

Q

R