DOCUMENT RESUME

ED 118 058 IR 002 895

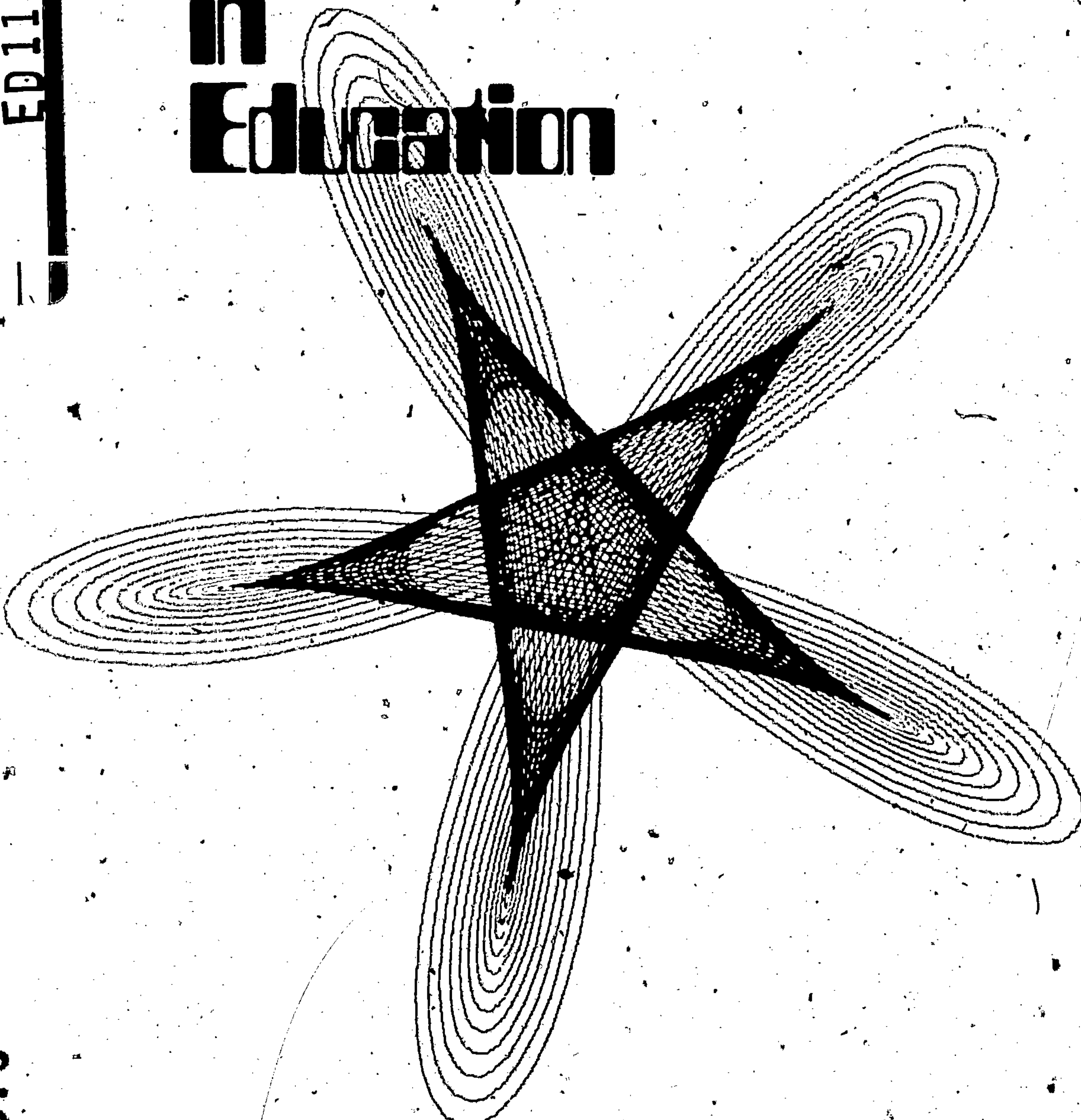
AUTHOR O'Neil, Carole; Pytlik, Mark
TITLE Computers in Education.
INSTITUTION Manitoba Dept. of Education, Winnipeg. Computer Services Branch.
PUB DATE [73]
NOTE 71p.

EDRS PRICE MF-$0.83 HC-$3.50 Plus Postage
DESCRIPTORS *Computer Assisted Instruction; Computer Oriented Programs; Computer Programs; *Computers; Computer Science; Computer Storage Devices; Data Processing; Educational Administration; Information Processing; Information Retrieval; Information Storage; *Manuals; Programing Languages; Time Sharing
IDENTIFIERS Canada; Computer Hardware; Computer Software

ABSTRACT

The Canadian Department of Education developed this manual to provide teachers and administrators with information about the potential use of computers. Part I describes at length the five components of the computer input, output, storage, control, and arithmetic/logic functions) and gives a discussion of computer languages, programing, batch processing, time sharing, and minicomputers. Part II covers a variety of administrative uses for computers. Part III lists the educational uses of computers, including computer-assisted instruction (CAI). A list of references, a 67-item bibliography, and a glossary of computer terms are included. (DS)

Computers in Education



2

Report of the

Computer Services Branch Department of Education Province of Manitoba

The illustration on the cover was drawn by a computer.

The points of view expressed in this report are those of the authors and do not necessarily reflect the opinions or policy of the Department of Education.

Foreword

Computers, with time, are continuing to have a significant impact on our lives. This impact is becoming evident in the field of education, especially in the areas of instruction and administration. Concomitant with the above trends there is a growing need for us to have access to relevant information on the possible services computers can render in education.

The Department of Education is pleased to make available this publication. Its intent is to provide teachers and administrators with information about potential uses of computers.

I am, therefore, pleased to recommend this report to all who may wish to learn more about computers and their services.

At this time I would like to acknowledge with appreciation the efforts of all those who participated in its preparation.

The Honourable Ben Hanuschak,
Minister of Education.

About the Authors

This report was prepared by Miss Carole O'Neil and Mr. Mark Pytlik as a summer project under the Student Employment Program.

Miss O'Neil received a Bachelor of Arts degree in Mathematics from the University of Winnipeg and a Master of Science degree in Computer Science from the University of Manitoba. She is currently teaching mathematics at Beausejour Senior School.

Mr. Pytlik received a Bachelor of Science degree in Mathematics from the University of Manitoba and is presently completing the requirements for a Master of Science degree in Computer Science at the same institution.

Preparation of the Report

This report was developed in three phases. In the first phase the authors interviewed thirty-one people to ascertain computer information needs of educators. In the second phase they conducted an extensive correspondence with educational institutions to receive information about their computer projects. In the third phase they researched available library resources and wrote the report.

Acknowledgements

The authors wish to thank all those who assisted in the preparation of this report. They would especially like to thank Mr. Peter Luba, Co-ordinator, Computer Services, Department of Education for his constant direction and encouragement.

TABLE OF CONTENTS

Introduction

Too often people think of computers as 'giant brains' capable of unlimited, awe-inspiring feats. Although it is true that the computer can perform computations incredibly quickly, it would be an idle, useless piece of machinery if there were no one to tell it what to do. The computer is, in fact, a 'high-speed idiot'. It cannot think. It must be told exactly what to do in a precise, step-by-step sequence of instructions formulated by a human being. The value of the computer lies in its tremendous speed and unfailing accuracy. Whereas human beings tend to become careless in performing tedious, repetitive tasks, a computer can perform its assigned tasks 24 hours a day, 7 days a week with no loss of efficiency.

The computer has extended our capacity to do intelligent work, enabling us to solve problems more quickly, with less drudgery, and with greater accuracy. The result is a much-improved ability to discover, create, build, solve and think that will affect how we study, work, and live our everyday lives.

Today computers play a key role in science, government and industry. Many banks use computers to speed the handling of cheques and automatically maintain bank accounts. Every major airline uses a computerized reservation system to maintain up-to-date information on all its flights and allow passengers to reserve seats within seconds by telephone. Man could not have landed on the moon without the computer's help to guide his spacecraft. In Toronto a computer controls nearly all the city's traffic lights to keep traffic flowing smoothly. The Federal Government is the largest user of computers in the country. Everyone is aware of at least one of its computer applications — the automatic processing of income tax returns. These are just a few of the ways computers are being used today. New computer applications are being developed daily and computers are fast becoming indispensible. There are at present over 4,000 computers in Canada. This number is expected to reach 20,000 by 1980 with a total investment in computer systems of 12 billion dollars. [1]

Computers have been with us for only a very short time — about 25 years — yet they have already had a profound effect on our lives. Probably no single machine has had as much influence on us, yet is more misunderstood, than the computer. Why is this so? Why do so many myths and misconceptions about computers prevail among the general public?

Probably the most widely held misconception about the computer is that it can think — that it has a mind of its own. The computer is only able to do complex computations and make simple decisions. Yet it could do neither of these if some person had not given it explicit instructions to cover any situation that might arise. Thus the computer is no more effective than the person telling it what to do.

Are computers dehumanizing? Are we being reduced to mere numbers for the convenience of the computer? Too often computers have been used in an unfeeling way, increasing our frustration and our suspicion of them. This need not be. With a minimum of extra effort the computer can be made to communicate with people in a more 'human' way.

Are computers putting people out of work? Although fewer people will be needed for routine tasks which are better done by computer, many new computer-related jobs have emerged. Extensive retraining will be required for people to take advantage of these opportunities.

Is the computer a threat to our freedom? With the advent of large computerized data banks, is vital information concerning our lives an open book to any person with access to these computer files? Even now, employers, banks, and government officials have access to personal information on us. This information may influence our credit rating or jeopardize our chances to get a particular job. Unless very strict rules are enforced as to who may access computer files, these fears may actually become a reality.

In this report we shall attempt to answer some of the questions the layman might ask about the computer and its role in education. What is a computer? What can it do? What can't it do? How does it work? How can it help the administrator? the teacher? the student? We will try to answer these questions in clear, non-technical language with a minimum of jargon and a maximum of examples.

Part I

How Computers Work

What is a Computer?

A computer is essentially any device for processing information. In this report we shall only consider the *electronic stored-program computer.* The operations of this type of computer are carried out electronically rather than mechanically. This computer also has the ability to store and retrieve information. It stores not only the data upon which it is to operate but also the set of instructions that it must follow to solve the problem. Whereas a calculating machine must be given new instructions at each step in a problem, an electronic computer performs each step automatically under the control of its stored 'program'.

Two terms are frequently encountered in any discussion of computers. One of these is *hardware.* It refers to the physical equipment and electronic circuitry that make up a computer. The other term is *software* which refers to the programs within the computer that make it operate effectively. These are the instructions written by human beings that tell the computer what to do. To understand computers we must have some knowledge of both hardware and software. Let us look at hardware first.

Five Components of a Computer

Every computer system — from the desk-top minicomputer to the large research model occupying an entire floor of a building — has five basic functional components. They are:

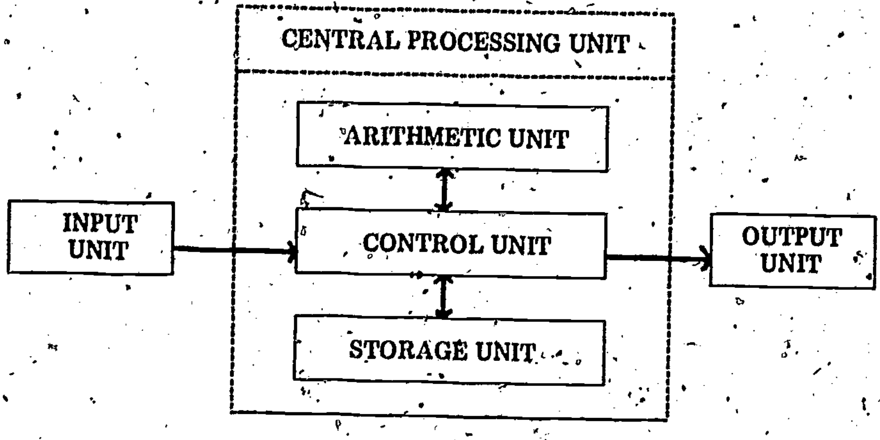
1. Input
2. Output
3. Arithmetic/Logic
4. Storage
5. Control.

They represent the five logically distinct tasks to be performed in a computer system, but a great variety of physical equipment could be chosen to accomplish each one.

The heart of a computer system is the *Central Processing Unit,* or *CPU* as it is usually called. It consists of three sections: storage, arithmetic/logic, and control. The *storage unit* ('memory') is the place where the data and programs are stored. The *arithmetic/logic unit* performs addition, subtraction, multiplication, division, and also logical operations such as comparisons. The *control unit* co-ordinates the activities of the other four units, and causes the instructions that make up a program to be executed.

The computer first needs to 'read in' the programs and data with which it will be working. This is accomplished by the *input unit.* When the results have been obtained they must be 'written out' by the *output unit* in a form which human beings can understand.

The relationship between these five components is illustrated by the following diagram:



We will now discuss each unit in more detail.

Input

Perhaps the most common form of input to a computer system is the punched card. The card is divided into 80 vertical columns each of which is used to represent a single character. Within each column there are 12 positions where holes can be punched in a coding structure known as Hollerith coding, after its inventor who first used it for the United States Census of 1890.

When all the information has been punched into cards by the use of a keypunch machine, the complete card deck is loaded into a *card reader.* The card reader can sense the position of the holes in each card electronically and send the resulting pulses to the Central Processing Unit (CPU). A typical card reader can process 500 cards per minute.

Another device operating on the same principle is the *paper tape reader.* Information can be coded as a series of holes on a continuous paper tape and sensed in the same way as the holes on punched cards.

Both of these devices operate very slowly compared to the speed at which the CPU can accept information. A more efficient input device is the *magnetic tape drive.* Data can be recorded on magnetic tape in the form of coded magnetized spots. The spots can be packed very closely on the tape — usually 800 characters per inch for a total of 20 million characters on a reel of tape.

All of the above input methods have a serious drawback. Using a machine with a typewriter-like keyboard, a typist transforms the information in the source document into holes in cards or paper tape, or to spots on magnetic tape. This operation of *data preparation* is time-consuming and prone to error.

What we need are methods of communication which are convenient to men, rather than to the computer. Surely the function of the computer is to cause less work for man, rather than more. One input device providing the desired flexibility is the *typewriter terminal*. It consists of a keyboard very similar to an ordinary electric typewriter plus the electronic circuitry to convert each keystroke into signals to the CPU. The user types his instructions and data and the computer prints its reply on the same sheet of paper. We will have more to say later in this report about the potential of this terminal for instruction.

There are also several types of input devices capable of sensing marks made directly on cards or paper. A card reader can be modified to sense pencil marks, instead of holes, on specially designed *OMR (Optical Mark Reader)* cards. These cards are particularly advantageous for student use since they eliminate the need for a keypunch. There are other devices frequently used for test scoring, which can read and process entire pages of pencil marks. Banks make widespread use of devices that read characters written in magnetic ink, such as on personal cheques. Finally, readers are available which process typewritten or even hand-written documents. The characters must be in predesignated areas and carefully formed. This form of direct character recognition is still a complex and expensive form of input.

Output

The reports and data generated by the CPU are most commonly conveyed to the user by a *line printer*. High speed printers are capable of printing up to 2,000 lines per minute on a continuous paper form. Special forms such as paycheques or invoices can be used if desired. If the output is to be used later as input for another program, it can be punched into cards by a *card punch*, or written onto magnetic tape by the same magnetic tape unit discussed earlier.

A *plotter* can be operated under computer control to produce charts, graphs, and drawings. The computer can use an *audio response unit* to construct spoken messages from a small vocabulary of pre-recorded words and phrases. For example, stock market quotations could be produced and sent out by telephone without human intervention.

We have already mentioned that a typewriter terminal can be used as an output device. However, it is noisy and rather slow. The *Cathode Ray Tube (CRT)* is an output device which overcomes these problems. A CRT terminal is similar in appearance to a television screen and can display characters, digits, lines, graphs, and so on. It also includes a keyboard for communicating with the computer. An additional feature available with some CRT's is a *light pen*. It can be used to point to a specific location on the screen — for example, to choose a response to a multiple choice question, or even to draw a graph. CRT's have the advantages of fast, quiet operation and excellent display capabilities. However, they do not provide a permanent record ('hard copy').

Storage

Internal Representation of Information

The Central Processing Unit consists essentially of a complex network of electronic circuits. An electronic circuit is a two-state device (either 'on' or 'off'), thus it can represent a digit in the binary system (base two). While our decimal system uses the digits 0 through 9, the binary system uses only the two *binary digits*, or *bits* symbolized by 0 and 1. Successive numbers from right to left represent increasing powers of 2 rather than powers of 10.

Thus the binary number 1110110 represents no units (2^0), one two (2^1), one four (2^2), no eights (2^3), one sixteen (2^4), one thirty-two (2^5), and one sixty-four (2^6). In decimal this is 118 — one hundred (10^2), one ten (10^1) and eight units (10^0).

Decimal	Binary
$= (1 \times 10^2) + (1 \times 10) + 8$	$= (1x2^6) + (1x2^5) + (1x2^4) + (0x2^3)$
$= 100 + 10 + 8$	$+ (1x2^2) + (1x2) + 0$
	$= 64 + 32 + 16 + 4 + 2$
	$= 118$

Arithmetic operations are in principle no different in either system, as in the following example of addition:

Decimal	Binary
118	1110110
25	11001
143	10001111

This notation is very conveniently represented by electronic circuitry and so is used by most computer manufacturers to store and manipulate arithmetic quantities inside the Central Processing Unit. Computers are also able to represent non-numeric information in terms of bits. There are a number of coding schemes which can represent the letters of the alphabet, special symbols such as $ and +, and the characters 0 through 9. Each character is assigned a unique combination of bits, and this coded information can be handled easily by the CPU.

The ability to store and retrieve information is what makes the computer something more than an ultra-fast calculating machine. Most computer applications do not involve complex scientific calculations ('number crunching') but rather the processing of large quantities of non-numeric information. Recent technological advances such as lasers, superconductivity, and micro-miniaturization of electronic circuits promise continuing increases in the speed and capacity of the computer's storage devices.

The instructions and data currently in use must be stored and retrieved very rapidly by the CPU. The type of storage used for this is called *main storage* and operates at very high speeds. The *access time* required to find any specified piece of information is measured in *microseconds*, (one microsecond = one millionth of a second) or even in *nanoseconds*, (one nanosecond = one thousandth of a microsecond). Computer systems require far more storage than main storage can provide, however, so there are also *auxiliary storage devices.* Auxiliary storage devices can hold much more information but access times are much slower, ranging from a *millisecond* (one thousandth of a second) to several minutes.

Main Storage

All the information held in main storage is encoded into combinations of 0 and 1 bits. Most computers use *core storage* to represent the bits. Large numbers of tiny doughnut-shaped cores made of magnetic material are strung on a grid of fine wires. The magnetism in each core can be in one of two states, and the wires are used to detect or change that state. Cores are grouped into *words* containing from 12 to 80 bits, depending on the computer model, and each word can be manipulated as a unit. Computer memories range in size from a few thousand to several million words, and constitute a major proportion of the computer's cost.

The important characteristic of main storage is that each word is 'addressable', that is, a unique numeric address is associated with each. The instructions to the computer refer to specific locations by number and enable us to read the information stored there any number of times without altering it. However, when we write information at a location the old value stored there is lost. For this reason the word 'memory', although frequently used, is inappropriate for describing the storage unit of a computer since only the most recent information is retained. This is another unfortunate anthropomorphic term, along with 'read', 'write', 'electronic brain', which causes people to erroneously ascribe human capabilities to an inanimate machine.

Auxiliary Storage

The magnetic tape drive, which was previously described, can also be used as an auxiliary storage device. Information is organized sequentially on the tape in groups of related data called *records.* For example, a record could consist of all the information about one student. To find a particular record on the tape requires that all records must pass the *read/write head* until the correct one is reached. This may take several minutes. Most business and administrative applications such as payrolls process the data in a fixed order, so this *sequential access* is convenient. In many other applications we want the capability to seek out a record directly without reading any others first. This is called *random* or *direct access.* Two direct access storage devices in common use are the *magnetic disk* and *magnetic drum.*

A *magnetic disk pack* consists of 6 or more disks coated with magnetic material and looks somewhat like a stack of phonograph records. Information is stored as magnetized spots arranged in concentric tracks on the surfaces of the disks. Each surface has its own read/write head on a moveable arm. A *disk drive* keeps the disk pack in constant rotation at a speed of about 1500 revolutions per minute. Information anywhere on the disk pack can be accessed within milliseconds by moving the read/write head to the correct track and waiting for the data to spin by.

The *magnetic drum* operates on similar principles but at even higher speeds. Information is stored in parallel tracks on the surface of a cylinder coated with magnetic material. As the drum spins it passes beneath fixed read/write heads. The amount of information on the drum is fairly small but it can be stored or retrieved very quickly.

In deciding which auxiliary storage device to use we must reach a compromise among three factors — cost, speed of access, and storage capacity. As in all things, you get what you pay for, so that the faster access speed you demand, the less capacity the device will have and the more it is going to cost. A medium or large-scale computer will usually have multiple tape, disk, and drum units while small systems will usually have only one type of auxiliary storage.

Control Unit

The control unit must supervise and co-ordinate the operation of the other four units of the computer. It directs the input unit to accept information and move it to the storage unit; it directs the output unit to fetch information from storage and display it on any requested output device. Under the direction of the control unit, the arithmetic/logic unit copies numbers from main storage, performs arithmetic operations on them, and stores the results for later use.

The sequence of instructions which the computer must follow to perform a specified task or to solve a problem is known as a *program* and is drawn up by a *computer programmer.* The instructions a computer can recognize directly make up its *machine language.* Each type of machine has a unique language.

The instructions making up a program are first loaded into main storage, and the control unit is given the location of the first instruction. The control unit then fetches each instruction from storage, decodes it, and causes it to be executed by issuing appropriate commands to the other four units. Ordinarily the instructions are executed in sequence but some instructions cause a branch or transfer to another location in the program. This permits subsets of the instructions to be coded once and then repeated as many times as necessary.

Arithmetic/Logic Unit

Every computer has a set of machine language instructions which can be executed by the circuitry in the arithmetic/logic unit. These pre-wired instructions include simple arithmetic operations on numbers, moving information to and from the storage unit, and comparing or testing two quantities. Each operation takes a few microseconds or less — thus a typical computer can perform one million operations per second. The number of built-in instructions ranges from a dozen to 200 or more depending upon the design of the machine. The computer is only able to tackle a problem whose solution can be reached by a series of these basic operations.

Computer Languages

In order to communicate with a computer, man must write his instructions in a language the computer can understand. Many such languages have been developed, each with its own grammar, punctuation and vocabulary. However, the only type of program the computer can execute *directly* is one written in its machine language. Programs written in any other language must be converted into equivalent machine language instructions by a program called a *translator.*

We will first discuss how programs are written in a machine language and then mention some of the widely used programming languages.

Machine Language

To illustrate the principles involved let us look at some simple machine language instructions for a hypothetical computer called SPECTRE.[2]

Every instruction has two parts — an *operation code* telling which operation is to be performed, and an *operand* which specifies a location in main storage to be used in the operation.

Operation Code → | CC | WWW | ← **Operand**

All arithmetic operations are carried out using a special location in the arithmetic/logic unit, called the *accumulator*. We assume our main storage has 1,000 locations numbered 000 to 999.

We will consider only six instructions:
(WWW is a number between 000 and 999)

Instruction	Meaning
10WWW	Clear the accumulator and copy into it the contents of storage location WWW.
11WWW	Copy the contents of the accumulator into storage location WWW.
20WWW	Add the contents of location WWW to the accumulator.
30WWW	Read a number from a card and store it at location WWW.
31WWW	Print the contents of location WWW.
40000	Stop (no operand required).

Here is how we would program the SPECTRE computer to read two numbers from punched cards, add them together, and print the result.

Instruction	Explanation
30187	Read a number and store it at location 187.
30188	Read the next number and store it at location 188.
10187	Clear the accumulator and copy the number at location 187.
20188	Add the number at location 188 to the accumulator.
11189	Copy the number in the accumulator into location 189.
31189	Print out the number at location 189.
40000	Stop.

As you can see programming in machine language could become tedious and time-consuming. Clerical errors are frequent since the programmer must keep track of the addresses where the data is stored (187, 188 and 189 in our example) and must be completely familiar with the numeric operation codes.

Assembly Language

People quickly realized that coding a large program such as a payroll entirely in machine language is an almost impossible task. For that reason simple languages called *assembly languages* were invented to let the computer take over most of the 'bookkeeping'. In an assembly language we substitute easily remembered letter combinations called *mnemonics* for the numeric operation codes and use symbolic names instead of numeric addresses. A special type of translator called an *assembler* can translate these symbolic instructions into machine instructions.

In the case of our hypothetical machine language we could use the following mnemonic operation codes:

Mnemonic Operation Code	Numeric Operation Code	Key
CLA	10	CLear and Add to the accumulator.
STO	11	STOre the accumulator.
ADD	20	ADD to the accumulator.
INP	30	INPut a number.
OUT	31	OUTput a number.
STP	40	SToP.

We also use symbolic names like A, B, SUM to refer to storage locations. Our assembly program for adding two numbers appears on the left, and the corresponding machine language instructions produced by the translator program appear on the right, as follows:

Assembly	Machine
INP A	30008
INP B	30009
CLA A	10008
ADD B	20009
STO SUM	11010
OUT SUM	31010
STP	40000

Assembly languages are much simpler to read and understand but they are still at a low level of sophistication and are generally used only by experienced programmers. Each machine instruction must have its assembly counterpart on a one-to-one basis, so there is no reduction in the number of instructions to be written. The programmer must still be very familiar with the detailed functioning of a particular computer and the entire program must be re-written for any other make or model.

These restrictions led to the development of 'high-level' languages which are easier to learn and more convenient to use, and are not dependent on the type of computer.

High-Level Languages

Each high-level language has a well defined set of rules, vocabulary, and punctuation that must be used in writing programs. To use this language on a particular computer, we must provide a translator program called a *compiler* to translate statements in the high-level language into the machine language of that computer. Compiler programs are usually provided by the manufacturer as part of the computer software.

There are two widely used programming languages which together account for perhaps 80% of all the programs ever written. The rules for these languages are well standardized and compilers for them are available on most general purpose computers.

The first of these is called FORTRAN (FORmula TRANslation) and was developed to aid scientists and engineers in mathematical computations. Its arithmetic statements closely resemble algebraic notation.

For example, the formula

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

becomes

X(1) = (—B + SQRT (B**2 — 4*A*C)) / (2 * A)

using * for multiplication, ** for exponentiation and / for division.

FORTRAN has extensive mathematical and scientific capabilities and is often the language taught in introductory computer courses.

COBOL (COmmon Business Oriented Language) is the standard language for commercial and business applications, such as payroll and accounting programs. Its statements are written in a subset of English and convey some meaning even to those not trained in the language. Consider a typical statement from a payroll program:

IF HOURS-WORKED IS GREATER THAN 40.0 THEN GO TO OVERTIME-SECTION.

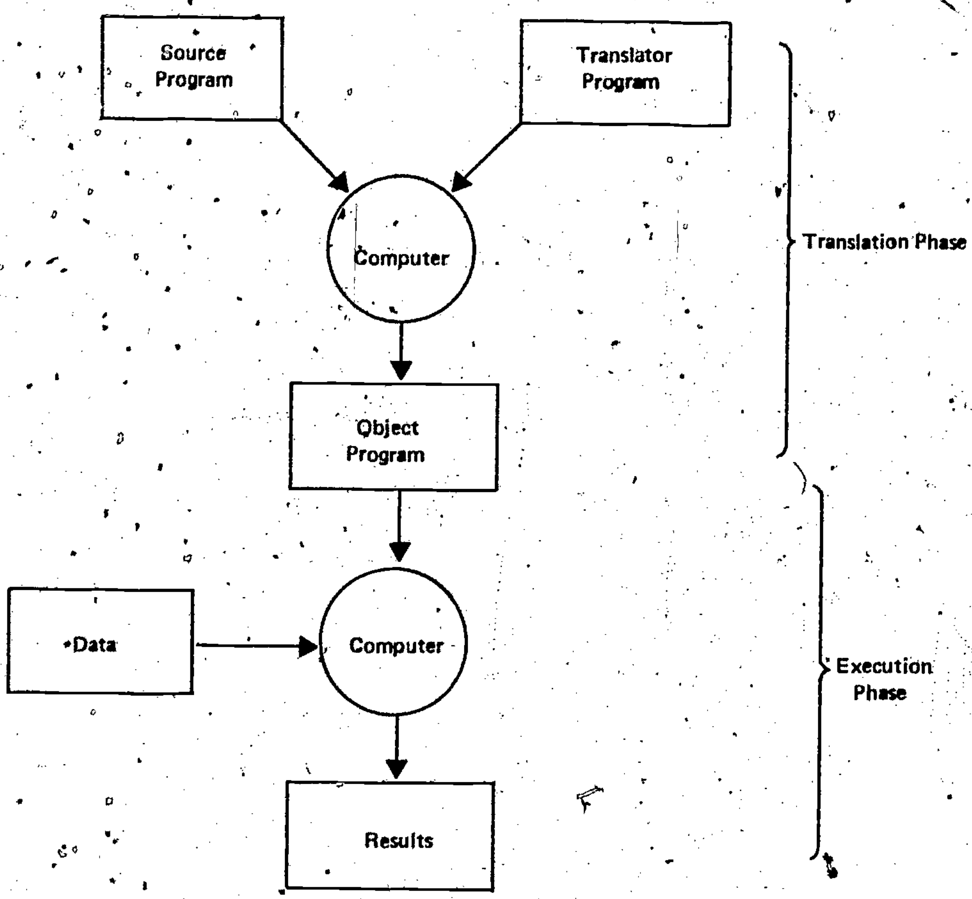
Less widely available is a language called PL/1 (Programming Language One), which combines some of the better features of FORTRAN and COBOL. It is the closest thing yet to a 'universal' language since it is equally suited to both business and scientific applications. It has some very powerful features; for this reason its compiler programs are extremely complex and not as frequently used.

Many other computer languages are in existence, but they are either for special purposes or else not widely distributed.

Running a Program

What happens when we 'run' a program on a computer?

The complete program is first punched into cards and loaded into the card reader. A special first card tells the computer what language is being used; for example FORTRAN. The computer must load the FORTRAN compiler program into main storage and then load the user's *source program* also. The compiler program now takes over and translates each FORTRAN statement into the equivalent machine language statements. As it does so the compiler also checks for errors in the source program and prints a diagnostic message for each error so the programmer can correct it. If no errors are found this *translation phase* ends, and the *execution phase* can begin. The compiler program and source program are no longer needed and only the machine language program — the *object program* — is kept in main storage. The instructions in the object program are now executed one at a time, and the computer reads in data when requested by the program and prints out the results. The following diagram shows what happens:



Preparing a Problem for Computer Solution

Not all problems are suitable for solution on a computer. We must first define the problem precisely and decide whether there is a detailed step-by-step procedure for finding the solution. Such a step-by-step procedure is known as an *algorithm*. We must specify each step in complete detail so that the computer never has any doubt about what to do next.

A large problem may include a number of complicated algorithms, and it is usually easier to represent each algorithm by a diagram called a *flowchart*. Flowcharts enable us to visualize the logical decisions involved in any kind of process. A flowchart makes it easier to spot errors in logic and it provides a convenient means of communicating the solution to others. A flowchart for an everyday task like running a bath is shown on the following page. Rectangular boxes represent actions, diamond-shaped boxes represent logical decisions to which the answer is either YES or NO, and the arrows show the logical sequence.

A computer can make 'logical decisions' — those whose answer is either YES or NO — but only by comparing two numeric values. We must be able to quantify all variables in the program — that is, assign numeric values — or else the computer cannot manipulate them at all.

After we are satisfied with our flowchart, the next step is to write a computer program using whatever programming language is most efficient and convenient. We must then run the program on the computer using test data to check that it works correctly in all possible situations The error checking facilities in the compiler will detect errors in the use of the language but cannot check for logical errors. If the program gives incorrect answers or if there are certain circumstances we forgot to allow for, then the program must be revised and rerun until all the 'bugs' are removed. This procedure is called *debugging*.

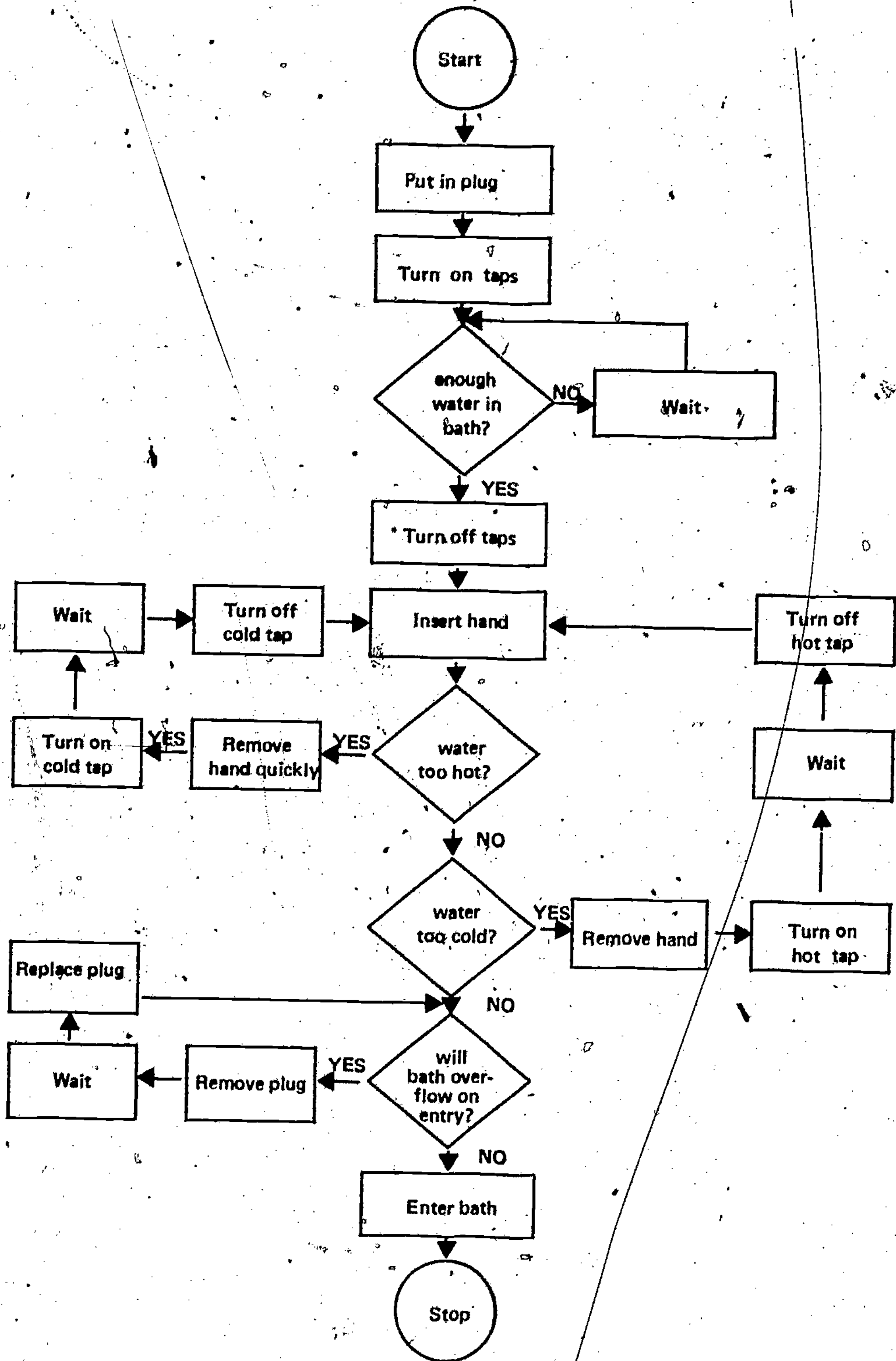
If the program is one we intend to keep and re-use frequently then we can *document* the program. This means we will have to write a set of notes describing the overall purpose of the program, the details of the algorithm, samples of input and output, and complete instructions on the use of the program. This information is indispensable if the program is to be modified at some later date. The importance of complete documentation especially for complex business applications cannot be overstressed.

Batch Processing

Most computer centres use what is known as *batch processing*. Programs to be run are grouped together and submitted to the computer to be processed consecutively without intervention by the computer operator. The computer automatically controls the transition from translation phase to execution phase for each job, and then from one job to the next. The correct compiler is provided automatically by the computer.

Batch processing generally has a card reader for input and a line printer for output; it is especially suited to large volumes of input/output. However, the time between handing in the cards to be run and finally receiving the

RUNNING A BATH



printed output can be quite long — from a few minutes to 24 hours or more — because all the jobs in the same batch must be finished before anyone gets any output. Jobs using a lot of computer time may be held back until the computer is less busy.

These restrictions can be tolerated in a business environment where a tight schedule is adhered to but can be very frustrating to a person developing a new program. Consider a programmer's frustration when he waits 24 hours only to find his program would not work because he left out a comma somewhere! It can easily take 10 or 20 such runs before his program works correctly.

Time-Sharing

These drawbacks motivated the development of an entirely new way of using computers — *time-sharing*. A large time-shared computer can serve one hundred people or more at the same time, each unaware that anyone else is using the computer. Each user has his own terminal — usually a typewriter terminal or a CRT terminal — which can be located at any distance from the computer and connected to it by ordinary telephone lines.

The person seated at a terminal types in a set of instructions; the central computer processes these instructions and types out a response within seconds; the user types further instructions or corrections depending on what answer he had received, and so on. The key concept is that this process is *interactive* — a continuous dialogue occurs between the user and the computer. The user can write a program at his own speed, run it, and within seconds, get the solution or error message(s). The inevitable errors can be corrected as he goes along and an entire program developed during one session at the terminal.

Time-sharing systems have another significant feature. The computer has a library for storing programs. Each user has a section of his own where he can save work-in-progress and any programs that he has developed. A large variety of library programs are also available to any user. These ordinarily include mathematical and statistical programs for which the user only supplies the data, and a number of games for recreation or entertainment. Computerized football games, card games, etc., help remove some of the fear and unfamiliarity with the computer which can intimidate the novice computer user.

How Time-Sharing Works

Time-sharing is made possible by the extremely fast speed of the Central Processing Unit — one million or more instructions per second. Suppose a time-sharing system has one hundred users. Then in every second the CPU could do several thousand arithmetic operations for each user and still have time left over. The CPU switches its attention rapidly from one user to the next, spending only a few milliseconds with each one. If a person's work has not been completed during his 'time-slice', the partial results are saved and then his work is resumed when it is his turn again. This round-robin method allows a large computer to service over 100 users concurrently yet the time each individual must wait at a terminal for a response is only a second or two. Time-sharing is not restricted to large computers. It can be performed by machines of all sizes down to economical minicomputers with only a few terminals.

New Languages

Traditional computer languages like FORTRAN and COBOL can be used on time-sharing systems, but new languages have been invented that take better advantage of the capabilities of interactive systems. These languages were specifically designed for the non-specialist and are very simple to learn and use.

Foremost among these is BASIC (Beginner's All-purpose Symbolic Instruction Code). It was developed at Dartmouth College, New Hampshire — one of the pioneers of time-sharing — with the stated aim of encouraging students and teachers to use the computer. The commands are in simple English and the language is easily learned by anyone willing to devote a few hours time. Some form of BASIC is available on virtually all time-sharing systems. Only a few elementary commands are required to write simple programs. A number of special features can be added on as the user finds need for them, making BASIC as effective as any other high-level language.

Another interactive language which has great potential is APL (A Programming Language). Its very compact and powerful notation allows one to write in 2 or 3 lines what might take 2 or 3 pages in another language. It is sufficiently simple that elementary and junior high school students in Edmonton have learned it successfully, yet powerful enough that the University of Laval has set up a complete computer system with 60 terminals which use nothing but APL.

Remote Computing

Advances in telecommunications now permit computer terminals to be located at any distance from the central computer. All that is required for this *remote computing* is a telephone. A person can dial the computer's number and his terminal will be connected by ordinary telephone lines to the computer. He will then have all the central computer's processing and storage facilities available to him. The user can be across the street or even across the country — however, long distance charges can be prohibitive. Because business use of remote computing is growing rapidly, telecommunications costs are being pushed down. Educational users will inevitably benefit from this trend. Cable television and eventually satellites may provide alternative low-cost communications for remote computing.

Time-sharing systems for educational purposes exist in many parts of the United States and Canada. One noteworthy example is the Dartmouth Time-Sharing System (DTSS)[3] centred at Dartmouth College, a small liberal arts college in New Hampshire. Dartmouth set out in 1963 to develop a time-sharing system with the intention of making the power of the computer available to all students. Today nearly 90% of all Dartmouth's students know how to program the computer. They use the BASIC language which was invented at Dartmouth. The campus has over 150 terminals located in 25 buildings. More than 50 colleges and secondary schools in or near New England have terminals and are regular users of the Dartmouth Time-Sharing System. They stretch from Montreal to Boston and from Maine to New Jersey.

The computer at Dartmouth is available as a resource in the same way the library is. A student may sit down at any of the conveniently available public terminals without asking permission and use it for whatever purpose he wishes. On a typical day, about 2,000 users 'log into' the system; in a year nearly 80% of the student body and 70% of the faculty make some use of it. The computer now pervades the entire curriculum. It is routine to have computer homework assignments in a wide variety of courses. More than half the computer's time is used for course assignments although there is a great deal of recreational use. The average undergraduate spends about an hour a week at a computer terminal. Dartmouth students have lost any fear or awe of the computer — it has become an everyday part of their lives. They are well prepared for a world in which computers play an increasingly important role.

Minicomputers

A significant trend in the computer industry in the past few years has been the rise of the minicomputer. Minicomputers now outnumber medium and large scale computers combined; meanwhile their cost continues to fall. Generally a minicomputer may consist only of a central processing unit that sells for $25,000 or less, or a complete system for $100,000 or less. The CPU for a minicomputer is about the size of a briefcase, yet it has a set of instructions and a speed comparing favourably with most computers available today. Minicomputers do not have as large memories as the big computers and they can't handle as many input/output devices. However, they perform well as a time-shared computer for instructional use.

The chief advantage of a minicomputer is its low initial cost. A basic minicomputer with one interactive terminal only costs about $5,000, and a system with 16 interactive terminals can be obtained for approximately ten times that amount. Since the minicomputer is not at some remote location it can be available 24 hours a day, 365 days a year. No special facilities such as air-conditioning, raised flooring or special power lines are required. A minicomputer is rugged and generally gives no more trouble than a TV set. No specially trained operators are required. The biggest plus for the user is the availability of simple conversational languages such as BASIC. Minicomputers are also expandable — more storage and more terminals can easily be added as the number of users increases. Students can get valuable 'hands-on' experience when the computer is located in their school.

The minicomputer has two fundamental restrictions. Because of its generally small memory the size of the programs that can be written is limited. However, most student-written programs are quite short and are not usually affected by this limitation. Secondly, fewer computer languages are available. In particular, COBOL is rarely available, and FORTRAN is slow and not very convenient to use. These are real handicaps but an interactive system is really more suited to the simple conversational languages anyway.

Thousands of schools in the United States and Canada are buying their own minicomputer. As prices continue to fall, this becomes an increasingly attractive way of bringing the computer into the classroom.

Part II

Administrative Use of Computers

Educational administration is a complex and demanding process which has as much to gain from computerization as any business organization. The use of computers in banking, insurance, transportation, manufacturing, and many other industries is firmly established and even indispensable. Both education and business have similar procedures such as payrolls, financial accounting, and inventory control which are eminently suited to computer processing. Special educational tasks such as student scheduling cannot borrow directly from business experience but can still benefit from the use of computer methods.

Advantages

What advantages can be realized from the use of computers in educational administration? The chief benefit is the great reduction in routine clerical tasks. Large-scale organizations including school districts generate and process a great deal of information. To do all of this manually could keep an army of clerks busy. The operations to be performed are usually of a simple step-by-step nature, highly repetitive, and requiring great precision. These are exactly the characteristics of a good problem for computer solution. The computer calculates extremely rapidly, has infinite patience, and when properly programmed will perform for years without error. Its speed and accuracy make it ideal for handling any organization's routine clerical tasks.

Another advantage is the computer's ability to tackle complex and painstaking assignments which otherwise could not be attempted. Administrators can request summaries and reports which would have taken months by manual methods yet which can be provided by the computer within minutes. By means of the computer the school executive can greatly increase the quality and quantity of the information available to him and at the same time be more selective in its use.

Data Processing

In order to achieve its goals any organization must base its decisions upon information, that is, *data*. The 'raw data' is a mass of facts and figures which must be manipulated according to precise rules of procedure to produce new, more desirable information. This is known as *data processing.* When mechanical or electro-mechanical equipment is used to reduce manual processing to a minimum, we refer to it as *automatic data processing.* If the data is mainly processed by a computer we call it *electronic data processing* (EDP). EDP is the common term for the computerized clerical operations widely used in business and industry. Let us take a closer look at some of these.

An Example of EDP

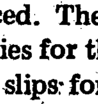
The first thing we need for any computer application is a *data file*. For example, in a payroll system we keep information about each employee such as his name, address, employee number, rate of pay, applicable payroll deductions, and year-to-date totals for gross pay, income tax, and Canada Pension Plan. This information constitutes a *record* and the collection of records for all employees forms a data file. This material changes relatively infrequently so we call it the *master file*.

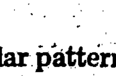
To calculate the payroll we also need the time cards, or other such record of attendance, for all the employees. This information constitutes a *transaction file*.

Both the master file and the transaction file are maintained in ascending order by employee number. These two files are the input to the payroll program.

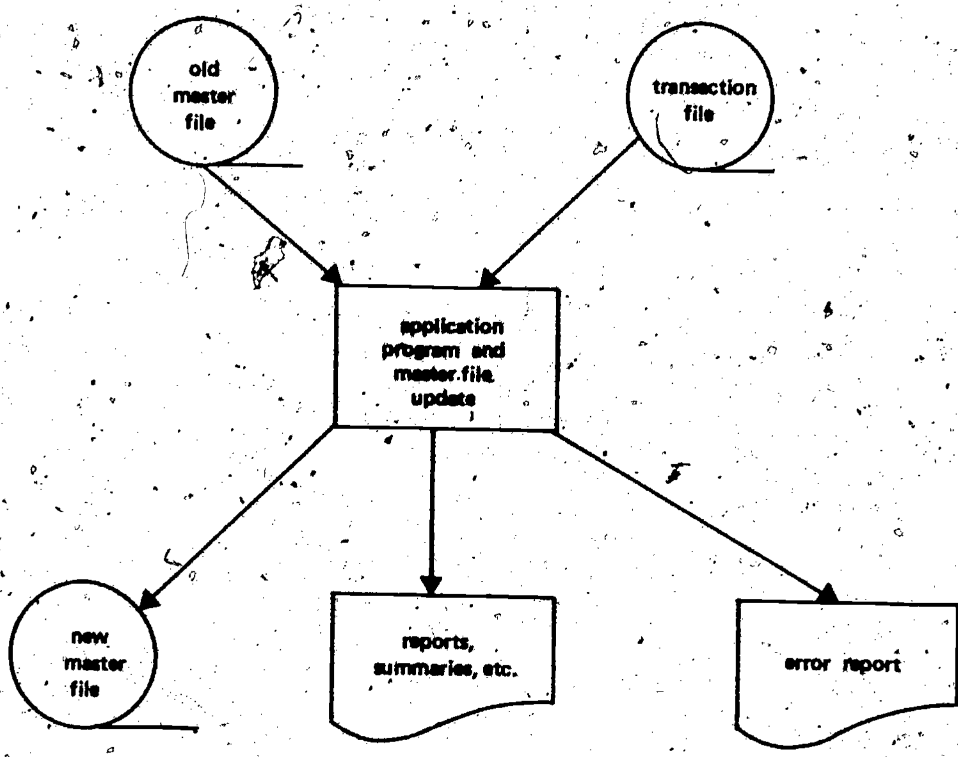
When the payroll program is run, the information on each transaction record is matched with the correct master record in order to calculate the pay for each person. This algorithm is straightforward but may be rather involved since it must take into account different rates of pay, undertime, overtime, and many possible deductions. A paycheque and statement of earnings is produced for each person. The master file has to be updated since year-to-date totals change. Also the transaction file might have included instructions to add new employees to the master file, delete retired employees, or change the information on a particular record, such as an address or rate of pay. All these changes must be incorporated into a new master file. Errors or omissions in the transaction file might be detected by the payroll program — for instance, time cards with an invalid employee number. In these cases an error report must be produced. The payroll program also produces a payroll register and other summaries for the use of management. A special annual run can produce all the T-4 slips for income tax purposes.

Every standard EDP application follows a similar pattern. Transactions are accumulated and then processed at regular intervals against the master file to produce required summaries and reports, an updated version of the master file, and an error report if necessary. The computer run might occur daily, weekly, bi-weekly, monthly, or even less frequently, depending on the requirements of the system.

The process is shown in the following flowchart, where 

represents a data file (usually on magnetic tape), and 

represents printed output.



Personnel System

Other elements of the personnel system besides payroll preparation can benefit from computerization. The employee data file may be expanded into a personnel file by including details of employment experience and qualifications. This data is useful in determining provincial grants, promotions, teacher placements, etc. Another file on retired teachers is helpful for pension plan administration.

A major component of the personnel system is the recruitment and selection of new staff. The computer can keep up-to-date lists of positions vacant and details on all applicants. A computer program can provide a list matching applicants to the available positions.

In large school systems the provision of substitute teachers can be a constant administrative problem. The Detroit Board of Education[5] makes creative use of a computer to help its 320 schools place an average of 500 substitute teachers a day — at a saving of more than $2000 a month over the previous system. The school representative telephones the computer between 6 a.m. and 9 a.m. if a substitute is needed that day. Using a touch-tone telephone or a dial phone with touch-tone pad attached, he keys in coded data to indicate his requirements. The computer searches its file of available substitutes and selects the most appropriate teacher based upon the subject to be taught, the teacher's credentials, and his distance from the school. Less than a second later the computer *verbally* responds with the directory number of the substitute. The caller then uses this number to look up the substitute's name and telephone number in a directory located in his school.

An accurate picture of the current financial status can be maintained and regular financial reports produced for management's use. Cost analysis is easier when the computer retains all the financial data. Previous years' budget data can easily be tabulated to aid next year's budget planning. Other computer programs can make projections about future trends, such as pupil enrolment, or can calculate the financial implications of alternative proposals, such as analyzing the costs of a proposed salary schedule.

Physical Resources Accounting

Two primary subsystems of physical resources accounting are inventory control and purchasing. Computers have proved very effective at maintaining records for thousands of items in inventory and indicating when to reorder each item. This tight control helps to minimize the cost of carrying inventory and to achieve savings through bulk purchasing. The master file includes information such as item number, description, location, quantity on hand and on order, cost, supplier, and information on issues and receipts. The inventory control system processes all requisitions and receipts against this file daily and so maintains an up-to-date inventory. Other programs analyze the activity for each item and optimize the maximum and minimum stock levels.

The purchasing system operates in conjunction with inventory control. When the re-order point for an item is reached, a report is produced and sent to the purchasing department, or the purchase orders could even be issued automatically, since all the necessary information is on file.

A school system could also use the computer to maintain facilities inventory. This would include a complete listing of the location and utilization of a school system's assets — classrooms, laboratories, gyms, etc. This information is essential if existing facilities are to be used to their maximum potential.

Maintenance and repair projects can also benefit from EDP. The planning, scheduling, and control of these projects can be automated to provide regular reports on completed and continuing work and on their costs. This kind of monitoring is especially useful in managing school building programs.

EDP may also improve the efficiency of the school transportation system. Computer programs have been developed for the routing of school buses to produce much more efficient routes — requiring fewer buses driving fewer miles and yet providing better service. The possible savings for rural school districts operating large bus systems can be substantial.

The computer applications reviewed above have been in the area of business administration and borrow heavily from EDP practices in private industry. Let's look at some applications developed especially for school administration.

Student Services

Student Records

Today's school systems must maintain extensive data on their students — past and present. This information is constantly referred to and updated as the student progresses through the school system. Traditionally, these student records have been kept in file folders. This approach presents major problems for large school districts, including lack of storage space, misfiling, and clerical time lost searching out information in the files.

The computer's ability to store and classify data and prepare reports can alleviate most of these problems. The information in the student master file can be maintained efficiently and effectively using EDP techniques. An important plus is that this file can be accessed by other computerized subsystems such as student scheduling, grade reporting, attendance, etc.

Student Scheduling

The increasing trend toward individualized programs of study for secondary school students has made the scheduling of school activities more complex than ever. The consolidation of small schools into larger units, the increase in course choices, modular scheduling, the need to maximize the use of facilities — all these have made the computer's assistance a virtual necessity in scheduling classes for a large school.

The first step is to tally the course requests for all the students. This will be greatly facilitated if students use course request cards of the OMR type which require no keypunching. The computer provides the administrator with a report of the number of students requesting each subject, so that he can decide how many sections are required. A potential conflict matrix is also produced, listing for each pair of subjects the number of students requesting both. This indicates the subjects which should not be scheduled at the same time.

The next step is to build a master timetable giving time, place, teacher, and class size for each course. The master timetable is usually built up by the administrator's taking into account special circumstances a computer couldn't know about. The administrator's task is much easier with the aid of the computer-produced course request listings and conflict matrix. Few computer scheduling systems generate their own master timetable because the program is very complex, generally requiring a large machine to produce a satisfactory timetable, and the cost is usually prohibitive. However, the

University of Laval has developed a promising and sophisticated system which produces at reasonable cost a fully optimized schedule for the entire university.[6]

The next step is master schedule simulation. The computer uses the tentative master timetable to assign as many students as possible to their requested classes. Conflicts and unsatisfied requests are printed out, along with the proposed teacher, student, and room schedules. The master timetable can then be adjusted by the administrator and a further trial-run undertaken. This procedure is repeated a number of times until an acceptable timetable has been produced for as many students as possible.

Finally the computer prints out the results, including teacher, student, and room timetables, class lists, and the final master schedule.

Computerized student scheduling is a very successful application of the computer's talents to a specialized educational problem. A Student Scheduling Service like that just described has been offered for a number of years by the Education Data Processing Branch of the Ontario Ministry of Education. In 1972 almost 280 schools in Ontario chose to use it.[7]

Grade Reporting

The production of pupil report cards requires substantial effort on the part of teachers. Much of this work can be automated — but not all. Teachers must prepare accurate information for the computer such as class mark lists by subject. Using this information a simple program can prepare individual report cards, honour rolls, failure lists, school marks summaries, and statistical analyses. Each school can specify its own layout for any report and receive as many copies as necessary. Grade reporting is a routine task for a computer and saves much human time and effort.

Attendance

For various legal and financial reasons our schools are required to collect and report statistics of students' absences and lateness. EDP techniques can be used to summarize student attendance data and generate the necessary reports. The production of detailed statistical analyses is a relatively simple programming task. The computer can even assist in the keeping of daily attendance. The teacher fills in the lates and absences on special forms with a lead pencil and the computer uses these to keep the register up to date. The method of recording data must be simple to use and relatively error-free. Otherwise more teacher time would be required than with the traditional procedures and no true saving would result.

Test Scoring

Automated test scoring and analysis is a well-known application of the computer in secondary education. The computer is well-suited to processing objective-type achievement and ability tests. Answers to the multiple-choice questions are filled in with a pencil and the answer sheets are read automatically by a device known as an optical scanner. The results are rapidly tabulated and pertinent statistics such as means, standard deviations, and correlations are calculated. Automated test scoring becomes invaluable when large numbers of students are involved.

Library Services

Many school libraries are using computers to help with their operations. Basically, five such operations can be managed by computer: cataloguing, ordering, reference, circulation, and financial control.

As a new book or other resource item enters a library which uses computer services, vital information such as call number, author, title, subject heading, publication data, and content description is added to the library master file. This material may be conveyed to the computer on punched cards or, more conveniently, by teletype. This information can then be used to produce catalogue cards as well as spine, pocket, and charge card labels. The master file can also be used for inventory and ordering purposes. If a required title is not found in the master file, the computer can actually print out a purchase order.

In many school libraries, the computer is being used to manage the 'charge-out' and 'charge-in' of library materials. The master file can be queried concerning the availability of a particular title and, if necessary, the requestor put on the waiting list for that book. At any time, the librarian can consult the master file to determine who has a particular book, when it was taken out, when it is due, how many times it has been renewed, and who has placed a 'hold' on the book. The computer can automatically issue overdue notices or check if a book is on the reserve list. Once circulation duties are taken over by the computer, library management statistics may be produced as a by-product. Such information as charge-out totals by user categories, requests for reserved books, and total circulation figures by subject areas can be listed. In addition, financial accounting reports can be generated by the computer.

In some large college and government libraries the entire reference system is maintained by computer. Rather than search for a reference in the card catalogue, the user can call for a display of that reference by the computer. If he makes a request by author or subject, he will receive all references in that category. Because this system is quite expensive and is only feasible for very large libraries, it will probably not appear in our public schools for some years to come.

Management Information Systems

The computer applications to educational administration discussed thus far operate relatively independently of each other. Each one has its own data file, resulting in a great deal of wasteful duplication of information between files. It is very difficult to use information from a number of separate files for the preparation of a report or for evaluation and research purposes — for example, a study relating student grades to the use of library resources. In addition to automating routine administrative tasks the computer can also be used to aid senior administrators in planning and policy making. There are ever-increasing pressures on our school systems to increase services, maximize effectiveness, and minimize costs. These pressures emphasize the need for a *management information system* to provide adequate, accurate, and timely information for educational decision making.

The characteristic feature of a management information system is the use of an *integrated data base.* This means that the school system maintains computerized files for all areas of educational information — facilities, finance, instructional programs, personnel, and students. Cross-references and linkages are established between all of these files. There is minimal duplication of information between files. Interrelated items can be retrieved from any of the files without a time-consuming search. Careful thought must be given to what data should be collected, from whom, how often, how it is to be recorded, and how it is to be used. There is an ever-present temptation to record something merely because it is available, rather than for its usefulness. The creation of an integrated data base of educational information is an ideal application of the computer's ability to store and retrieve data.

The integrated data base, once established, is used to support all the information processing activities of the school system. This includes the regular EDP applications discussed previously — payroll, accounting, scheduling, etc. But more importantly, information is now available on a regular basis to assist senior administrators to judge the effectiveness of present programs and to plan for the future. Reports can be generated by the use of a management-by-exception technique where the computer flashes a warning when it detects a situation beyond pre-set limits. Reports can be distributed on a 'need-to-know' basis so that middle management gets the detailed information it needs while senior officials are freed of administrative details and get only a broad outline.

Many advanced management techniques can be used with a management information system because of its inherent flexibility. It can provide the raw data required by such techniques as PERT (Program Evaluation and Review Technique), CPM (Critical Path Method), and PPBES (Planning, Programming, Budgeting, Evaluation System). These methods allow evaluation of alternative courses of action and allocation of scarce resources to meet specified objectives.

An example of a successful management information system is the TIES Project [8] sponsored by 30 Minnesota school districts located in the Minneapolis - St. Paul area. TIES stands for Total Information for Educational Systems. A total of nearly 230,000 students and 35,000 employees are served by TIES. Participating school districts range in size from 1,400 students to over 31,000. Administrative, instructional, and research services are available to the member districts for $6.25 per student per year.

TIES uses an integrated data base and an on-line telecommunications network for maintaining it. Each district administrative centre accesses and maintains its own information files using a CRT terminal connected by telephone to the computer centre. Information can be displayed, added or corrected within five to seven seconds. This information is used by the regular administrative systems which include a Financial Budget and Accounting System, and a Payroll Personnel System. Special reports can be generated as easily as routine reports, without extensive redesign or reprogramming, thanks to the advanced design of the TIES system.

Extensive in-service workshops are a key factor in helping school personnel make use of TIES services.

TIES also supports the full range of instructional computer uses. Their instructional network has approximately 200 teletype terminals and 15 card reader terminals available for use by students and teachers. In some districts all students are able to write programs in the BASIC language by the end of the fifth grade. However, students are generally not introduced to the computer until junior high school. They first use the computer for simulations and then write their own programs using the simple, English-like statements of the BASIC language. The computer is most commonly used for problem solving in subject areas such as science, mathematics and social studies. Experimental programs are also underway in computer-assisted and computer-managed instruction.

TIES represents a dramatic change from current educational practice. It has proved one of the most successful uses of the computer in education.

Who Provides Computer Services?

There are a number of ways a school division may obtain administrative computer services. The first way is to either purchase or lease its own computer. Generally there is less problem in replacing leased equipment should more powerful equipment be desired at a later date. When a school division operates its own computer centre it must hire experienced staff to use the computer to best advantage. The staff of the centre usually first implement EDP procedures such as payroll and accounting, which show an immediate financial return. Larger centres can provide the whole range of administrative services, but each service requires careful planning and it may take several years before it is fully operational.

A second way of obtaining computer services is from a commercial service company. Many companies sell specific services such as payroll or scheduling to school divisions. A division can purchase the services it needs and can specify deadlines and penalty clauses in the contract. However, the services available may not truly meet requirements; there is little freedom to try new approaches; and a full range of services would be quite expensive.

Another alternative is the regional consortium. A number of school districts can cooperate to provide low-cost computer services to their members through a single large computer centre. This brings the cost of computer services even within the range of small divisions. This approach has been tried successfully in the United States. One example is the TIES project which was previously discussed. Another regional consortium is the Region IV Education Service Centre [9] in Houston, Texas which provides computer services for 700 schools and half-a-million students. One benefit of these consortia is that their large multi-purpose computers can provide a broad range of instructional services as well as education data processing services. A regional consortium on this scale requires a great deal of time and money to develop.

Another approach is to have the provincial Department of Education make administrative computer services available to all school districts. This is the approach being taken by the Ministry of Education in Ontario through its Education Data Processing Branch.[10] This branch has the computer resources and experienced staff to develop administrative services and to offer them to local school boards at cost. If a school board feels it can obtain more cost-effective services elsewhere or wants to use its own computer it is free to do so. Many local boards link their smaller computer via telephone lines to the large EDPB computer in Toronto and use its extra power when necessary. The provision of computer services to all school boards facilitates standardization in reporting, and more effective administration of the educational system by the Ministry of Education.

We have said little about the way application programs are developed. A recent survey[11] of large American school systems has shown that school districts with their own computer have a decided tendency to use their own staff to develop administrative computer programs. Contract staff are rarely used and existing application programs from other locations are used only occasionally. Yet many application programs vary little from one place to another and are usually available for the asking. Perhaps one reason for this lack of sharing is that most programs have not been written in a way that permits easy modifications to suit other school divisions' requirements. Another is the fact that no clearinghouse exists to assist in the exchange of application programs, and so most computer centres continue this wasteful duplication of effort. No matter who develops the computer applications a key point in their successful utilization is full explanation of the service to those who will use it and careful training in the procedures to be followed. Suspicion or misunderstanding can render even the most advanced and sophisticated computer application ineffective.

Part III

Educational Use of Computers

Each year, more and more schools throughout Canada and the United States are using computers for instructional activities. Considering the cost involved, why are these machines becoming so popular? The answer must be that the computer is producing significant improvements in the learning process. Let us briefly note some of the reasons why computers are becoming widely accepted in education:

1. The computer is an excellent motivational device. Most students like to work with the computer — to have it do work for them — to play games with it.
2. Since computers affect many areas of our lives, it is to the student's advantage to become acquainted with what they can and cannot do. Students can learn this by using the computer and by studying about it in class.
3. Because the computer can perform complex computations very quickly, it can help the student with lengthy calculations which threaten to obscure the learning situation. In fact, the student can use the computer to help him solve many problems which might otherwise be too complex to attempt.
4. Using the computer, a student can experiment with a model of a real situation, studying the outcomes as he varies any of the controlling factors. This technique, known as simulation, is especially used in the subject areas of science, social studies, and business.
5. The computer can assist in the instructional process by helping the teacher to drill, review, teach, or test students. This type of assistance is known as Computer-Assisted Instruction.

Let us look at how computers can be used in the classroom.

Learning About Computers

A large proportion of schools in Canada and the United States offer instruction in at least one of the following categories: computer literacy, computer science, and data processing.

The goal of *computer literacy* is to give all students basic information about the computer and how it affects their lives. Such information need not constitute a separate course, but may be part of the social studies, mathematics, or business education programs, and may begin as early as the junior high level. Topics such as the following could be discussed:

- a brief history of computers
- how computers work
- the capabilities and limitations of computers
- how computers are involved in our lives
- the misuse of computers.

One of the best ways to learn about the computer is to write programs for it. After learning a simple programming language such as BASIC, the student can write a few simple problems and run them on the computer. If a computer is not available, the student could experiment with a simulated computer such as CARDIAC, a small cardboard computer developed by the Bell Telephone Laboratory.

Computer science courses involve a more in-depth study of the computer and programming. The following is a list of suggested topics:

1. The components of a computer and their functions.
2. The study of algorithmic thinking and flowcharting.
3. The fundamentals of programming, including input, output, branching and subprograms. The student should learn both a low-level language (either assembly or machine) and a high-level language such as FORTRAN, BASIC, or PL/1.
4. Discussion of computer applications in such areas as industry, business data processing, transportation, communications, science, medicine, and recreation.
5. Some understanding of compilers, assemblers, operating systems, monitors and other kinds of software.

More and more business, industrial and government organizations are using computers for data processing. A course in *data processing* would generally be offered by the school's vocational or business education departments. Some topics which could be included follow:

1. The study of unit record equipment (e.g. sorters, interpreters, collectors, tabulators, etc.).
2. Components of the computer and their functions.
3. An introduction to computer programming in both a scientific language (e.g. FORTRAN, PL/1) and a data processing language (e.g. COBOL, RPG).
4. Computer applications to such business problems as payroll, etc.

The only equipment needed for the computer literacy or computer science courses is an interactive terminal or a remote batch terminal. If the latter is used, either a keypunch machine or OMR cards will be necessary. It is preferable that whatever equipment is used be readily accessible to all students. In addition to a terminal, data processing students would also benefit from hands-on experience with unit record equipment.

The Computer as an Aid in Instruction

In this section, we will look at some of the ways teachers and students are using the computer to facilitate learning. We will consider examples from many subject areas to demonstrate the breadth of possible applications.

There are basically two ways in which students can use the computer — *problem solving* and *simulation*. In problem solving the student usually writes his own computer program, and in so doing gains a deeper understanding of the algorithm upon which it is based. Alternatively, in simulation he uses pre-written (or 'canned') programs and only supplies data and observes the results. Obviously the former use requires a greater knowledge of programming and takes more time. In choosing whether to provide a canned program or have the students write their own, the teacher must decide whether studying the algorithm or studying the results is the main objective of the assignment.

At all times the temptation to use the computer only for the sake of using the computer must be avoided. The computer should not simply duplicate what a teacher can do in a classroom or laboratory, but should enhance with some additional dimension. It should not be used solely for jobs which could be done just as easily by a desk calculator — there is the danger of having the computer looked upon merely as a 'giant calculator'.

Problem Solving

Perhaps the simplest way of using the computer for problem solving is as a calculator. The computer can allow the class to look at more relevant, realistic problems by itself performing those time-consuming, error-prone calculations between the setting up of a problem and the arrival at its answer.

Let us first examine the subject of mathematics which best lends itself to computational aid. Can you remember having to find the roots of a quadratic using the quadratic formula? Computations could become quite tedious unless 'rigged' equations were selected. Although the most important step in the solution was the correct application of the quadratic formula, most of the time taken in arriving at the answer occurred after this step. Since performing the calculations does not enhance the understanding of the quadratic formula, we could have the computer do this work for us.

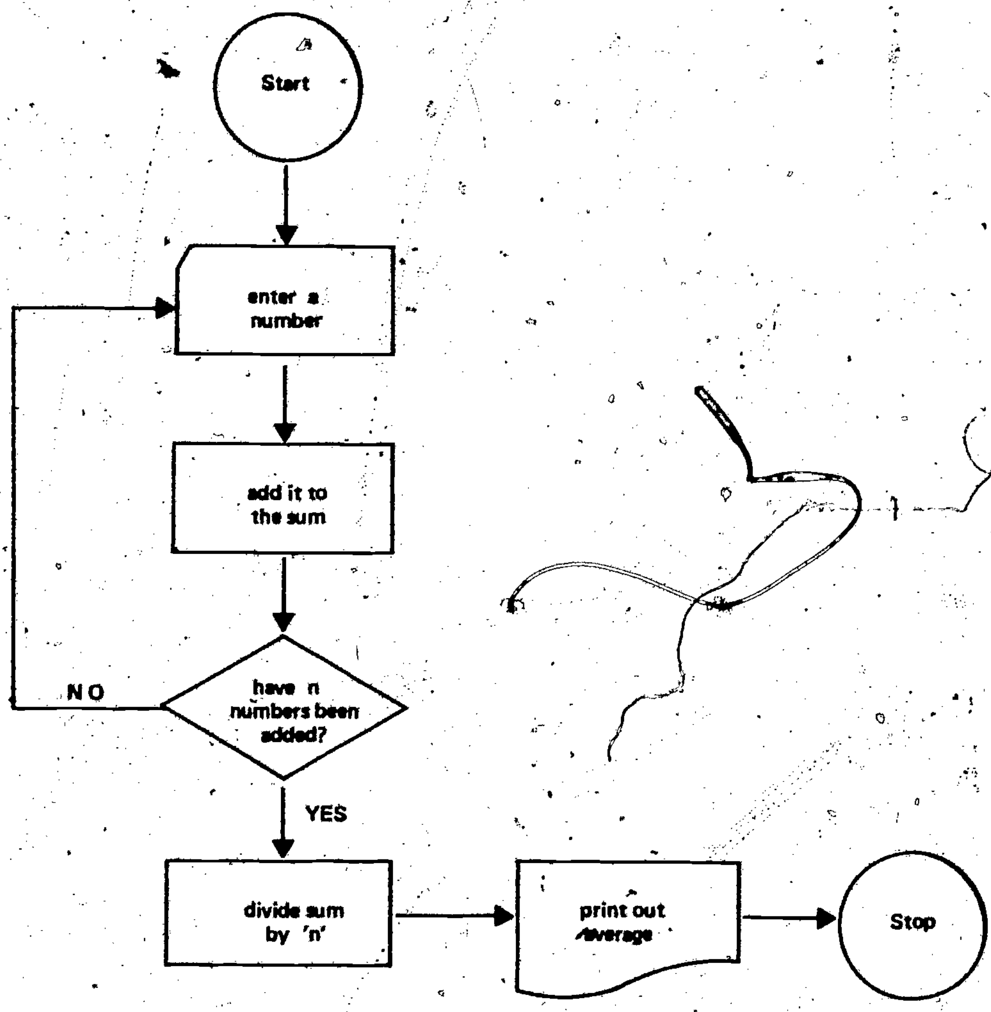
By having the computer evaluate a function closer and closer to a chosen point, the concept of 'limit' could be made much more concrete and meaningful to the student. The fact that this would require numerous difficult calculations is no deterrent to the computer.

Physical sciences, such as chemistry and physics can also benefit from the computational powers of the computer. For example, after performing experiments in the lab, students may have the computer analyze their data for them rather than doing it by hand.

When using the computer as a calculator we relegate to it most of the tedious calculations. Another reason for using the computer for problem solving is to reinforce concept understanding. The student must make a step-by-step logical analysis of the problem. In so doing, he acquires a deeper understanding of the nature of the algorithm, its power, and any potential difficulties. The writing of a computer program to solve a problem is more beneficial than working through two or three 'messy' examples by hand.

We could look at the process of programming as 'teaching' the computer how to solve the problem. As an example, suppose our problem is to find the average of an arbitrary number of integers. The first step is to decide how to attack the problem — in this case, we add all the integers and then divide by the number of integers. We next design a flowchart to help us conceptualize the problem. This process eliminates many errors and false assumptions and clarifies vague understanding. Our flowchart would look something like the following:

FINDING THE AVERAGE OF n NUMBERS

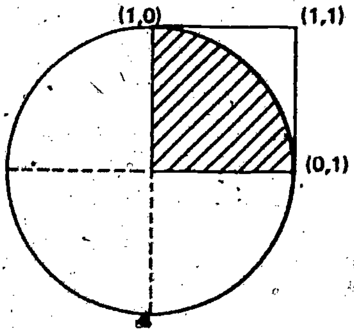


The hard part is over — it only remains to translate the flowchart into computer instructions and run the program.

The process of algorithmic thinking, very important in the sciences and other subject areas, is reinforced by the writing of flowcharts and computer programs.

What are some of the ways computer problem solving is being incorporated into the curriculum? Again, the subject of mathematics stands out. It is generally the mathematics teacher who introduces students to the computer. One well-known attempt to bring the computer into the mathematics curriculum is the CAMP Project[12] (Computer Assisted Mathematics Program), developed at the University of Minnesota. The CAMP series, which covers grades 7 through 12, was initiated to acquaint students with the problem-solving aspects of the computer in their regular school work. Thus, wherever the writing of a computer program would improve learning, the student is instructed to write one. Some of the topics which receive computer attention (selected from the grade 8 course) include: fractions, decimals, and percents; squares and square roots; properties of the rational numbers; geometric concepts: pi, perimeter, area, volume, properties of triangles, Pythogorean theorem, and so on.

An example of solving a standard problem by using the computer is evaluating the area of a geometric figure or the area under a curve. To calculate the area of a circle by computer,



i) we could set off a rectangle which includes the area to be calculated. (Let us choose ¼ of the circle).

ii) have the computer generate many pairs of random numbers whose co-ordinates are between 0 and 1. Count the number of pairs which lie inside and outside the required figure.

iii) then the number of points inside the figure, divided by the total number of points tried, times the area of the rectangle (which is 1 in this case), times 4 (since we have considered only ¼ of the circle), will approximate the area.

Note that the more points we generate, the closer our approximation will be to the actual area. This method would be impossible without the use of the computer, yet is probably more meaningful to the student than applying the formula $A = \pi r^2$. In fact, this approach not only reinforces the area concept, but could be used to develop the formula in a more meaningful way (by finding the areas of circles with different radii and comparing the results).

Outside the area of mathematics, subjects such as general science, chemistry, and physics may make use of computer problem solving techniques. Such topics as the study of the interrelationship between volume-temperature-pressure of gases or mass-volume-density of solids could be studied in detail by the computer. An inventive teacher and his students can discover virtually unlimited computer applications in these fields of study.

In order to solve problems using the computer, the student must have some degree of expertise in at least one programming language. The type of system the school employs could be either interactive or batch. If the system is interactive, the student will use an interactive language such as BASIC or APL; if batch, he will use a language such as FORTRAN or PL/1.

Problem solving is the most widely used application of the computer in instruction. For one thing, relatively little terminal or computer time is required — once the teacher assigns a problem, the student analyzes it and writes his program, using the computer only briefly to test and debug his program. As was mentioned previously, more complex problems may be attempted with the aid of the computer. Also, the student can try more examples because of the speed of the computer in performing calculations.

Simulation

The use of computers for simulation is one of the most attractive applications — and one which is suited to a wide variety of subject areas. In a *simulation*, a model or representation of a real world phenomenon is created, and its characteristics may be explored by experimenting with the model. Generally, little knowledge of the computer or how to program it is necessary — the student is expected to supply and juggle variables and observe the outcome. Applications range from generating Bach-like music scores, to performing a complex chemistry experiment, to landing a lunar module. If the student is proficient in programming, he may enjoy creating simulation programs of his own.

When should computer simulations be employed? Professor Ludwig Braun[13] of the Huntington Computer Project, has suggested several criteria:

1. When facilities or equipment needed to conduct an actual experiment are costly or complex and, as a consequence, where the experiment would probably not otherwise be performed (e.g. complex chemistry experiments).
2. When the actual experiment is hazardous and might endanger the experimenter (e.g. science experiments which involve radiation, high temperatures, explosive gases).
3. When time scales involved are either too short to allow easy measurement or too long to fit into the school year (e.g. biological studies in genetics, — observing successive generations of a particular species).
4. When the sample size available in the real world is too small to permit generalizations (e.g. the study of rare diseases by medical students).

5. When the experimental technique is complex and must be developed over an extended period of time.

6. When it is impossible to experiment directly (e.g. studies of political, economic, and social systems, human genetics).

Where do simulation programs come from? To develop realistic, accurate simulation programs, a team of system engineers, computer scientists, and subject matter specialists generally work together. Less complex simulations may be created by individual teachers or students.

Let us consider some specific simulation programs in use today, selected from a wide range of subject areas.

In the field of social studies, a simulation program called BALPAY [14] (internation BALance of PAYments) has been developed for the REACT (Relevant Educational Applications of Computer Technology) series. In BALPAY, the student plays the role of decision maker for a hypothetical country. He attempts to make economic decisions which will give the country a, healthy balance-of-payments position. The objective of the program is to give the student the chance to investigate personally the relationships between policy decisions and their ultimate effects on the economy.

A scientific simulation in widespread use is LMLAND[15] (developed for Project SOLO) which simulates a lunar landing. The lander has 30,000 units of fuel and starts its descent at 70 miles. The student, as the pilot of the module, can fire his thrust engines to brake his descent. One unit of fuel is used for every 100 pounds of thrust per second. He must land his craft at less than 30 feet per second to be considered successful. The program is realistic in its simulation of gravitational pull on the craft and thrust produced by the engines.

WHEELS[16] is a simulation developed by the TIES Project. Its purpose is to reinforce the concepts of personal finance and auto ownership. The student may specify the type of car he wishes to buy and how he plans to finance it. He must worry about insurance, operating costs, and major breakdowns. The computer simulates the amount of usage of the car, major repairs, accidents and unexpected events for each month of operation.

In the field of biology, a program has been developed for the PLATO IV system which helps to teach the laws of inheritance in genetics.[17] The computer simulates for each student a stock of parent fruit flies — some normal, some with assorted mutations. When the student requests that a mating be made, within seconds the offspring are displayed on a device attached to the computer. The student experiments with several generations of fruit flies keeping in mind that mutant characteristics are generally recessive and may show up in future generations. This type of experiment would not be attempted in a normal biology class, yet with the aid of a computer simulation, several generations may be examined in a matter of hours.

A final example, in the area of ecology, is the POLUT[18] program developed for the Huntington II Project. POLUT simulates the effects of certain variables on the quality of a water resource. The student can vary any of the following:

— type of body of water (large pond, large lake, slow-moving or fast-moving river).
— water temperature.
— type of waste released into the water (industrial or sewage).
— rate of dumping waste.
— type of waste treatment.

After the student enters his variables, the computer will determine the oxygen content and the waste content of the water for each day, until the oxygen and waste contents remain constant. The student can select his output to appear in the form of a table or a graph or both.

This simulation allows students to "investigate the effects of variables, compare various pollution control strategies, examine hypothetical situations, make and test hypotheses, and predict the implications of certain scientific and economic decisions". POLUT should increase the student's understanding of and insight into the ecological problem of water pollution.

Computer-Assisted Instruction

Individualizing instruction is perhaps the most immediate concern of educators at the present time. Modern computer technology can help provide individualized instruction for every student. Let us refer to this computer-aided tutorial approach to teaching as *Computer-Assisted Instruction*, or more simply *CAI*.

The majority of existing CAI programs are in the areas of mathematics, languages and the physical sciences. It appears that not every subject field or in fact every area in a field is equally suited to computer-assisted instruction.[19] The course designer must decide when learning experiences could be brought about just as effectively and less expensively with alternate teaching approaches.

Let us examine the three types of CAI, namely drill and practice, tutorial, and dialogue.

Drill and Practice System

The drill and practice type of instruction is the least complex of the three to program. The computer is used only to supplement regular classroom instruction, to drill and review students on previously taught material. Each student is guided through a completely individualized program of drill tailored to his particular ability and rate of progress.

Generally the student works sequentially through a program, answering questions posed by the computer. A temporary detour may be caused by an incorrect response, but after clearing up the difficulty, the computer returns to the main program flow.

Let us look at a simple drill sequence selected from an elementary mathematics program on multiplication.[20] Each student response has been underlined.

4 X 6 = ?

24

GOOD

4 X 9 = ?

36

CORRECT

4 X 7 = ?

30

WRONG, TRY AGAIN

4 X 7 = ?

32

WRONG, TRY ONE MORE TIME

4 X 7 = ?

27

NO, 4 X 7 = 28

REMEMBER NOW 4 X 7 = 28

4 X 7 = ?

28

RIGHT

4 X 8 = ?

32

4 X 5 = ?

20

GOOD

CAREFUL NOW

4 X 7 = ?

28

VERY GOOD

Notice that the computer recognizes and responds to student answers. The student is immediately told whether or not his answer is correct and then, depending on his answer, he is either advanced to the next problem, given a second chance to answer correctly, or supplied with the correct answer.

Frequently, a time limit is imposed on the student — if he does not respond within a fixed length of time, the computer acts as though he had answered incorrectly.

The structure of a drill and practice program may be fairly complex if it is made to handle remedial branching, evaluation, and review. Stanford

University, for example, offers a series of review units in elementary school mathematics[21] which utilize computerized drill and practice. A computer-administered pretest determines the initial level of drill exercises to be presented to each student. After each subsequent lesson in the unit, the student's performance is evaluated and he is accordingly directed to a lesson of greater, the same, or lesser difficulty. Therefore, several 'streams' of varying degrees of difficulty must be programmed into the instructional unit so that the computer can adjust to individual differences.

Let us look at some of the reasons why drill and practice CAI systems are gaining acceptance:

1. The student works at his own rate on problems specially selected for his level of ability.
2. Since the student is interacting with an impersonal machine, he is more likely to try a questionable response. Fear of ridicule by his peers is eliminated.
3. The student's every response is immediately evaluated and reinforced.
4. Drill and practice systems generally provide for score-keeping, so that information on a student's progress is readily available to the student and to his teacher.
5. Some drill and practice routines offer the additional feature of analyzing student responses. For example:[22]

   WHAT IS 2 X 3?

   5

   LOOK CAREFULLY AT THE OPERATION. TRY AGAIN.

   5

   REMEMBER YOU ARE BEING ASKED TO MULTIPLY,
   NOT ADD. TRY AGAIN.

   6

   GOOD WORK!

In this program, the computer was programmed to anticipate reasonable errors and give the student hints, based on the nature of the error.

Tutorial System

In a tutorial system the computer, rather than the classroom teacher, assumes primary responsibility for teaching new material. The interaction is intended to approximate that between a patient tutor and his student. The teacher is no longer the sole vehicle of instruction — he has time to individualize his own instructional efforts.

Early tutorial systems were of the *linear* type — that is, every student was forced to follow a fixed path through the programmed material with the only variation being in his rate of progress. This method is poorly suited for CAI since it does not take into account the computer's evaluation and decision capabilities. A *branching* type of program is more acceptable since it provides one or more alternate routes to the same learning objective. The path a student follows depends on his previous responses.

Early CAI programs used questions of the *multiple choice* type. The student did not have complete freedom of response but chose from among 3 to 5 pre-selected answers. The limitations of this approach led to the development of programs which allow student-constructed responses.

An example of a tutorial program which allows student-constructed responses is the following portion of an introductory physics course develped at the University of Illinois for the PLATO IV sytem.[23] Student responses are underlined.

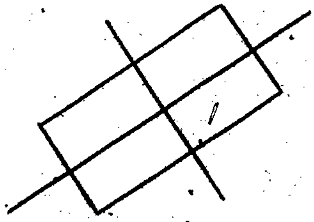
> If the acceleration is constant, the average velocity $\bar{V}$ can be written as a simple function of the initial velocity V_i and the final velocity V_f. Write an expression involving V_i and V_f:
>
> $\bar{V} = \underline{(V_f - V_i)\ /2}$ no
>
> Your expression gives the wrong result. Press — next — to see why.
>
> Consider a car that speeds up (with constant acceleration) from 60 to 80 fps to pass a truck. What would you say is the average speed $\bar{V}$ during this passing maneuver . . .
>
> $\bar{V} = 70$ ok fps
>
> Right but your formula gives
>
> $(V_f - V_i)/2 = 10.0$
>
> So you must re-write your expression.
>
> If the acceleration is constant, the average velocity $\bar{V}$ can be written as a simple function of the initial velocity V_i and the final velocity V_f. Write an expression involving V_i and V_f:
>
> $V = \underline{(V_f^2 - V_i^2)\cdot/2\,(V_f - V_i)}$ ok
>
> Fine. A simpler form is $(V_i + V_f)\ /2$.

A remarkable feature of this program is the ability of the computer to recognize the complicated expression as being correct and equivalent to the given simpler one.

Not only written information can be presented in a CAI tutorial system. With the aid of visual display devices, interactions such as the following are also possible (taken from a PLATO IV geometry course[24]):

a.

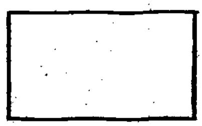
Draw a quadrilateral with only two lines of symmetry



Good! Your figure has symmetry lines that do not go through vertices.

b.

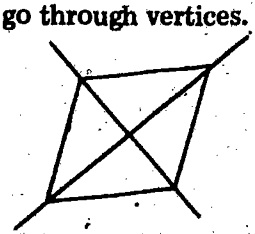
Now draw a quadrilateral with only two lines of symmetry that go through vertices.



You drew that figure before. Are you trying to fool me?

c.

Now draw a quadrilateral with only two lines of symmetry that go through vertices.



Very good!

Notice how the computer chooses an appropriate second problem (b) depending on the first response (a). Note also that the computer is able to classify the figures independent of size, shape or orientation.

Another feature which should be mentioned is the ability of the computer to perform computations during the instructional sequence. The following example from a Simon Fraser University chemistry course[25] will illustrate this. '*****' precedes each student answer.

Calculate the pH of a solution which has the following concentration of H (+):

1.0 X 10**-6 (Note: You may use CALC from now on
again, hope you remember how!)

*****calc log (1.0) + log (10** -6)

-6

*****-6

No, your sign is wrong, remember that pH = -log (H(+)).
Okay, try again.

*****6

Yes, that's good.

By typing the word 'CALC' followed by the expression (1.0) + log (10**-6), the student is temporarily removed from 'CAI mode'. The expression is evaluated by the CALC program and the anwer -6 is returned to the student's terminal. After the computer has printed the answer, the system is returned to the 'CAI mode' and is ready to receive an input from the student. Thus the student is able to work through fairly complicated problems without becoming bogged down performing lengthy calculations by hand.

Dialogue System

At the third and most complex level of computer-assisted instruction are programs which permit the computer to carry on a genuine dialogue with the student. Such systems, which are still in the developmental stage, are difficult to design since student responses need not appear as simple right or wrong answers. The computer must accept a student's response and, using some criteria, determine whether it is satisfactory. Also, student-initiated inquiries or comments must be interpreted and dealt with satisfactorily. The ultimate goal toward which researchers are striving is a dialogue system where the computer is able to recognize the student's spoken words and respond audibly in turn.

In the following excerpt from a program called ELIZA[26] developed at the Massachusetts Institute of Technology, the computer interacts as a psychotherapist might. Although this program is not strictly education-oriented, it is an excellent example of how a computer can be programmed to simulate a human conversation. The lines printed in capital letters represent computer responses.

Men are all alike.
IN WHAT WAY
They're always bugging us about something or other.
CAN YOU THINK OF A SPECIFIC EXAMPLE
Well, my boyfriend made me come here.
YOUR BOYFRIEND MADE YOU COME HERE
He says I'm depressed much of the time.

I AM SORRY TO HEAR YOU ARE DEPRESSED
It's true, I am unhappy.
DO YOU THINK COMING HERE WILL HELP YOU NOT TO BE UNHAPPY
I need some help, that much seems certain.
WHAT WOULD IT MEAN TO YOU IF YOU GOT SOME HELP
Perhaps I could learn to get along with my mother.
TELL ME MORE ABOUT YOUR FAMILY
My mother takes care of me.
WHO ELSE IN YOUR FAMILY TAKES CARE OF YOU
My father.
YOUR FATHER
You are like my father in some ways.
WHAT RESEMBLANCE DO YOU SEE
You are not very aggressive but I think you don't want me to notice that.
WHAT MAKES YOU THINK I AM NOT VERY AGGRESSIVE
You don't argue with me.
WHY DO YOU THINK I DON'T ARGUE WITH YOU
You are afraid of me.
DOES IT PLEASE YOU TO BELIEVE I AM AFRAID OF YOU
My father is afraid of everybody.
WHAT ELSE COMES TO MIND WHEN YOU THINK OF YOUR FATHER
Bullies.
DOES THAT HAVE ANYTHING TO DO WITH THE FACT THAT YOUR BOYFRIEND MADE YOU COME HERE

Although this program does not allow complete freedom of interaction between computer and student, it obviously provides for more than is possible with the tutorial method. One must keep in mind that this dialogue (or Socratic, as it is sometimes called) type of CAI is in its early stages and that the next few years should see some great improvements in the human-like quality of the interaction.

Some characteristics of ordinary dialogue which designers would like to incorporate into a computer dialogue system are:[27]

1. The computer can take into account not only the current question but also the previous portions of the conversations.
2. The user has the freedom of making any response, even if it is irrelevant.
3. The computer itself always responds with something relevant.
4. The computer can decide whether to delay information requested by the student.

5. Computer answers may be based on complex computations.
6. Verbal interactions should take place in common English.
7. Questions or interactions are permissible by the computer or the student at any time.
8. Non-verbal exchanges can include tables, graphs, pictures and sound.

CAI Hardware

Computer-assisted instruction is generally administered to several students simultaneously via terminals connected to a large time-shared computer. Although the computer is not physically capable of serving more than one terminal at any given instant, it can keep many busy while giving each student the impression that he alone is interacting with the computer. This illusion is accomplished by having the computer allow only short segments of time to all the active terminals in turn. Because of the rapid speed of the computer, it can check a student's response, or initiate the presentation of new material during one of these 'time-slices'.

There are a number of interactive terminal devices which could be adopted for CAI use. Unfortunately, most were not designed for students and have specific limitations.

In choosing a good CAI terminal one should keep in mind the following requirements:

1. The terminal should be easy for students to operate.
2. It should respond quickly and not keep the students waiting.
3. Its alphabetic and graphic display capabilities should suit the material to be presented.
4. It should not be subject to frequent breakdowns.

The National Research Council of Canada has prepared specifications for the design of CAI terminals and has developed a number of experimental models.[28]

The most commonly used CAI terminals are the *typewriter terminal* and the *Cathode Ray Tube* (CRT) display. The typewriter terminal, which is similar in appearance to an ordinary electric typewriter, is capable of displaying printed text. If a large amount of text is to be typed out at a terminal by the computer, an unsatisfactory length of time may be required. The CRT, which resembles a small television screen, is capable of displaying information much more quickly although no paper copy is produced. Generally, a keyboard is attached to the CRT to allow students to communicate with the computer. Another device which the student might use for this purpose is a *light pen*. The student need only touch the appropriate area on the CRT screen with the pen and his response will be recorded. This feature makes computer instruction possible even for very young children. The CRT can dispaly not only printed text but also tables, graphs and diagrams. Because CRT's are silent in operation, they may be installed directly in the classroom; whereas the installation of the noisier typewriter terminals would require a more specialized arrangement. A student instructional station may also include a computer-controlled slide projector and an audio unit complete with earphones so that the computer can actually 'speak' to the student.

Learning by ear, beginning with early childhood, is important and fundamental. However, experiments with audio devices are only in the developmental stages. In developing an audio unit the following standards should be met:

1. The speech should be of high quality and intelligibility.
2. Large quantities of possible messages must be available so that a message can be selected on the basis of the student's response.
3. Fast access to the messages is needed, in order not to delay the pace of the teaching.

The development of devices which recognize the human voice as input to the computer is still in the future.

CAI Software

An instructional program is composed of a series of *frames* — each frame consists of a teaching statement and some request for response from the student. The computer must be instructed to wait while the student is allowed time to respond. Regulatory and branching instructions, including the above 'wait' instruction, must be embedded directly into the programmed instructional sequence. These special instructions, which should be invisible to the student, are part of the CAI *authoring languages* which the course designer uses to write instructional programs. Authoring languages may be specially designed for CAI course writing (e.g. IBM's Coursewriter or CDC's Tutor) or may simply be multi-purpose interactive languages (e.g. APL). The language used must facilitate the rapid handling of large amounts of instructional material.

The author must also provide for error branching. That is, he must tell the computer how to handle any student response. How this is accomplished depends on the specific authoring language.

Finally, the completed CAI program must be tested on a number of students since it is impossible for the designer to anticipate all student reactions to materials which have been developed.

There has been much controversy over who should design CAI programs. Some believe that program development should be in the hands of individual teachers, while others maintain that teams consisting of experts in the subject field, psychologists, and experienced CAI programmers are necessary. Both approaches are presently being studied — the PLATO IV system and the TICCIT system described in the following section are examples of the respective approaches.

Some Current CAI Programs

Probably the most advanced and widely publicized experiment in CAI is the PLATO IV project[29] at the *University of Illinois.* PLATO (Programmed Logic for Automated Teaching Operations) was begun in 1960, with a single interactive terminal. The system presently supports 250 terminals located in schools and colleges in the Chicago area. Eventually 4,000 terminals will be included.

The most innovative feature of the PLATO system is the plasma display panel developed at the University. The plasma panel offers all of the capabilities of a CRT screen, but is very compact, has little or no flicker, and can be connected to low grade telephone lines. Computer controlled auxiliary equipment such as audio systems, touch panels, film projectors, and tape recorders, may be added to the PLATO IV terminal. Designers predict that the total cost of the PLATO system, including costs of the computer, communication lines, terminal equipment, and instructional material development, will be between 50¢ and 75¢ per student hour per terminal.

CAI programs are written in TUTOR, a powerful authoring language which can be used by teachers with little knowledge of computers. In fact, it is the individual classroom teachers who have developed the many instructional programs now on the PLATO system. Programs exist in such diverse areas as Biology, Chemistry, Computer Science, English, Foreign Languages, Geography, Mathematics, Music, and Physics. Lesson material can easily be revised by the teacher if he wishes.

Another system which has received much attention is the TICCIT (Time-shared Interactive Computer-Controlled Information Television) system,[31] developed by the Mitre Corporation. Unlike the PLATO system which depends on one large computer, TICCIT is built around two minicomputers. Interactive CAI terminals may be connected either directly to the system or remotely over cable television channels — less expensive than the traditional telephone line transmission. Each terminal consists of a colour television receiver, headphones, and a keyboard.

In contrast to the PLATO system, instructional programs for the TICCIT system are designed by teams of technical and educational specialists. It is estimated that with the present 128 terminals in the TICCIT system, the cost per student per hour should be about 35c.

Having looked at the two most outstanding computer-assisted instruction projects in the United States, let us look briefly at some projects under way in Canada.

Simon Fraser University[31] has been experimenting with CAI since 1969 when their Computer Centre implemented IBM's authoring language, Coursewriter III. In the same year, several interactive terminals were placed in two British Columbia schools, with the bulk of CAI programs being written in the field of science. However, most of the developmental work has been done at the university level in introductory chemistry. Generally, a lecturer and an experienced CAI programmer combine efforts to produce the instructional programs. Other CAI programs developed at Simon Fraser are in the subject areas of Physics, Mathematics, Biology, and Economics.

The Computer Applications department of the *Ontario Institute for Studies in Education* (OISE),[32] has been working for several years on a CAI remedial mathematics program for students with deficient mathematical skills entering colleges of applied arts and technology in Ontario. At present, about 600 students are participating in the computer-assisted course. The institute has also designed CAI courses in second language instruction and in high school physics.

For over five years, the Division of Educational Research Services at the *University of Alberta (Edmonton)*[33] has been carrying out research in CAI on their IBM 1500. They have developed a number of instructional programs, including APL for statistics labs, introduction to Coursewriter programming for teachers, pharmacology, nursing, mathematics, physics, fundamentals of data processing, and introduction to special education. The Faculty of Medicine is presently the largest user of the system for course work, particularly in the field of cardiology.

At present, a limited amount of experimentation with APL has been done in Edmonton public schools. Elementary students have successfully used drill exercises, while junior and senior high students have learned APL and used it for exploring mathematics.

The Quebec Department of Education[34] has also been experimenting with computer-assisted instruction. They have recently completed a three year project on CAI using an IBM 1500 educational system. Programs were developed for selected topics in French, mathematics, data processing, geography, and computer languages.

Some Problems with CAI

1. The greatest drawback to widespread CAI acceptance is the cost — terminal cost, computer cost, communications cost, and program development cost. The first two are at present quite high but technological advances are resulting in cheaper and better hardware. Transmission costs, too, should decrease — lowered costs have been observed in both the PLATO and TICCIT projects where CAI programs are transmitted over television channels rather than the usual telephone lines. Perhaps the largest cost and the one least likely to be reduced with modern technology is the development of instructional programs. The amount of time required to produce even a very short sequence is prohibitive. It is estimated that between 20 and 100 man-hours are required to produce one hour of tutorial CAI.[35]
2. There is still an acute shortage of high quality CAI course material. In fact, the programs that do exist generally cannot be used on systems other than the type on which they were developed. The shortage is also a result of a lack of professional and educational incentives, a lack of established production methods and procedures, and the vagueness of market prospects once programs are completed.
3. Teachers may resist CAI if they see it as a threat to their jobs or to traditional teaching methods, or if it requires considerable retraining. In order for CAI to be widely accepted teachers must find their new role attractive.

4. The relationship between student and computer lacks personal, human qualities such as the ability to listen to a student's voice, to observe his bewilderment, or to sympathize with his problems.

5. A number of educators are not convinced of the effectiveness of CAI as an instructional technique. As published reports on the effectiveness of present CAI systems become more widespread, this skepticism should decrease.

Some Praises of CAI

1. Computers make it possible to offer individualized instruction to a large number of students simultaneously. Consequently the student may work much more nearly at his own pace than in a regular classroom.
2. Much of the teacher's routine classroom work such as drilling, reviewing, and presenting straightforward material can be transferred to the computer.
3. Once an instructional program has been developed it can be used by many students and for many years.
4. Computers can continuously record and provide information on the achievement and progress of any student proceeding through a CAI course.
5. Since each student works individually, he feels free to make mistakes without fear of public embarassment.
6. Correcting mistakes and misconceptions immediately after a student has made them reinforces and improves a student's learning process.
7. The computer is impartial and consistent — it has no favourites. It never becomes impatient with a student's lack of progress.

Computer Managed Instruction

Many educators have been concerned that our school system is increasingly impersonal and that instruction is decreasing in effectiveness. They attribute this to increasing class sizes, to teaching for 'average' students at the expense of those at either extreme, and to a lack of contact between teachers and students. One reason for this is that teachers simply do not have enough time to handle the clerical work involved in classroom management and at the same time plan individualized learning programs for their students. Many educators now look to the computer for assistance. The computer has long since proven to be excellent at clerical tasks such as collecting, storing, retrieving, computing, and reporting. These functions find frequent use in the educational setting. The computer can help the teacher by storing test results, attendance, and permanent student records. It may also help him by analyzing the progress of each student and by continuously individualizing his instructional materials and assignments. Those tasks which the teacher can best do, such as introducing and discussing new concepts, explaining subtle points, and responding to student questions, he should continue to do. The routine clerical tasks which the computer can best do, it should be required to do. Thus, in no way can the computer be thought of as replacing the teacher — it is simply a tool which allows the teacher to spend more time with individual students.

In CAI actual course materials were stored in the computer and the student interacted directly with the computer. In contrast, in a *Computer Managed Instruction* system (abbreviated to CMI), the computer assists the teacher with all those activities which are involved in managing instruction, but it does not actually do any teaching itself. The computer basically performs two functions in a CMI system — it monitors the student's individual progress; it assists in planning his individualized program. Cooley and Glaser describe the goal of a CMI system as follows:

> The primary function of the computer in a CMI system is to make possible more complicated decision processes than would be possible without the computer and to do this on a continuous basis. Automation cannot be justified if the computer is used simply to keep records. Clerks tend to be cheaper record keepers than computers. In an individualized system, the teacher continuously needs information and assistance in making instructional decisions.[36]

There are typically four activities which may be managed by a CMI system. These are test scoring, diagnosing, prescribing, and reporting. Let us look at how a CMI system handles these processes.

Each course under computer management is divided into segments or instructional units. At the beginning of each unit, the student is given a *pre-test* which may be administered by either the teacher or the computer. In some CMI systems, a collection of test questions may be stored in the computer. The teacher may then compose his own tests from this collection and even have the computer print out enough copies for each of his students. If the tests are to be marked by the computer, the student's answers should be either on OMR (optical mark reader) cards, or on special answer sheets which can be processed by the computer. The purpose of the pre-test is to determine the student's initial status and direct him to a specific learning task. The computer may assign learning tasks automatically by examining test scores (this process is called *prescription)*, or if full automation is not desired, the computer can supply the test results in the form of a printed report to the teacher *(diagnosis)*. In this case, the report would become only one of a number of sources used by the teacher to prescribe learning tasks for the student. In the end, actual subject material may be presented through regular classroom instruction, reading assignments, programmed texts, taped lectures or even CAI programs.

Throughout the unit the student may be required to write *diagnostic* or *progress tests* to determine whether he is progressing satisfactorily. These again could be processed by the computer. When a student has completed the assigned tasks, he takes a *post-test* covering the unit. If he fails to reach acceptable performance, he may be assigned remedial material.

After the completion of each test, the teacher would receive several *reports*. One might list the name of each student, the unit of instruction, the objective, and the score for each objective in the unit. With this information, the teacher can study the pattern of accomplishment of each of his students and determine which of them require additional help. Another report might indicate how many students are achieving satisfactorily in each of the objectives for the unit. The teacher can use this information to detect common strengths and weaknesses in his class and adjust the instruction accordingly. The teacher might also request statistical information such as percentage scores, means, medians, modes, distribution, grade point averages, and percentile ranks to be calculated and printed out by the computer.

The basic pattern of pre-test, diagnosis, prescription, post-test and reporting would be repeated for each unit in the course.

Most CMI systems[37] are similar to what we have described above. Some do not have all of the above-mentioned features; others have incorporated additional capabilities. Besides providing the basic CMI services described above, Project PLAN (Program for Learning in Accordance with Needs) of the Westinghouse Learning Corporation plans a study program for each student and also provides career guidance. CAI is being incorporated as a method of instruction at the University of Pittsburgh by IPI/MIS (Individually Prescribed Instruction/Management Information System). AIMS (Automated Instructional Management System), developed at the New York Institute of Technology, includes evaluation of student progress, prescription, and empirical validation and optimization of instruction. At the University of Alberta in Edmonton a CMI system known as TAIM (Teacher-Authored-Instruction Manager)[38] is in operation. TAIM stores teacher-prepared lessons, tests, and decision algorithms; retrieves specific lessons and tests for individual students according to their needs; automatically scores multiple choice and numerical answer tests; permits rapid retrieval of specific information such as test scores; and allows easy modifications of lessons, tests, and decision algorithms.

CMI is gaining in popularity for a number of reasons. The cost of a CMI system is low compared to a CAI sytem, since CMI requires only batch processing. The student need not be on-line to the computer. Immediate response is not essential in a CMI system as long as results are obtained within a few hours after the data is submitted. Compared to CAI, CMI does not require much computer storage since lessons are generally not stored in the computer. In a CMI system, the computer acts as a servant to the teacher, performing chores which the teacher does not have time to do. With this type of assistance, the teacher has more time to work personally with his students.

Guidance Information Systems

Because the computer is capable of storing and quickly retrieving vast quantities of data, it can be used by the guidance counsellor to great advantage. It is especially helpful in the area of career guidance, where the counsellor would like to have information on thousands of careers and post-secondary educational institutions at his fingertips. A number of career guidance systems are now in operation, notably CVIS (Computerized Vocational Information System) in Illinois, Project PLAN in California, ECES (Educational and Career Exploration System) in Michigan,[39] and SGIS (Student Guidance Information Service) in Ontario.[40]

The simplest computerized guidance information systems require that students send their requests and personal data to a computer centre remote from the school. The student could request a list of colleges, occupations, information on financial aid. The list would be selected by the computer on the basis of student-supplied information such as his preferences, grades, financial situation, and so on. Requests from many students are processed in a batch and results are usually returned by mail to the student. There can be a considerable delay from the time the student submits his request to when he receives his reply.

Let us look at Ontario's SGIS which has only been in operation since February, 1972. At present, some 65,000 students in 75 Ontario secondary schools are using the SGIS service. The system presently contains information on 7,000 occupations and 200 educational institutions. There are four areas in which students may receive information: careers, educational institutions, career educational requirements, courses and programs. Each student is initially given a master list of institutions and careers and a supply of OMR (optical mark reader) cards. Students encode on these cards their specific requests and the cards are sent away to be processed. The information generated by the computer consists of a description of careers or institutions selected by the computer on the basis of the student's requests. Since the information is processed off-line, the student must wait from 3 to 6 days for his output.

This type of system has at least two drawbacks. One is the time delay involved. The other is the rigidity of the system. The student is not able to interact with the computer, change his mind, alter the course of the output — without submitting a new OMR card. If a student is undecided as to the career he should pursue or the institution he should attend, he needs a more flexible system at his disposal — one which allows students to communicate via interactive terminals with the computer. Such a system is being piloted in several schools in the province of Ontario in 1973. The student enters his requests using a CRT terminal. He can actively control the presentation of information during his session at the terminal.

Neither of the above systems stores information about the student, either personalized data or previous interaction with the system, for future reference. Some systems possess this additional feature. They maximize the computer's ability to relate a student's grades, interests, and abilities to his exploration of vocational and educational opportunities. The system may be designed to point out discrepancies in the student's planning, to review the student's record with him, and to review his last use of the system. Let us look at part of a sample dialogue from Project CVIS[41] at Willowbrook High School in Villa Park, Illinois which illustrates many of these features:

COMPUTER: Hi, Bill Brown. That is your name, isn't it?
STUDENT: Yes.

COMPUTER: The last time you were here, you were exploring occupations in the category of science that require community college training. What would you like to do today? Your wish is my command.

1. Continue exploration of this list of occupations.
2. Make new decisions about training.
3. Make new decisions about area of interest.

STUDENT: 3.

COMPUTER: Please choose one of the following (displays list of eight categories of interest).

STUDENT: Technology.

COMPUTER: You showed significant interest in this field on the last interest inventory you took.

A computerized guidance system can never replace the guidance counsellor. The counsellor is still needed to discuss individual decisions with students. The computer helps with the information-collecting and sorting stage previous to actual career decision making.

Conclusion

The intent of this report has been to make it possible for educators to learn more about how computers can be used in education.

The first part — How Computers Work — showed that despite its complex technology the computer is based on simple principles. Two main areas of computer use were then described — administrative and instructional. Most administrative applications of electronic data processing characteristically use the computer to greatly reduce routine clerical tasks and increase the efficiency of operations. The sophisticated management information system helps school administrators in all phases of planning and decision making.

Instructional uses of computers have progressed beyond the experimental stage. Computer problem solving and simulation are now well-established in thousands of North American schools and colleges. Computer-assisted instruction has been shown to be a good way of individualizing learning. Factors which inhibit its widespread acceptance are not so much technological as psychological and economic.

As computers become smaller, more powerful, and less expensive, they will play an increasingly important role in education. A population of teachers and administrators who are well-informed about computers is the best assurance that computers will be used efficiently and effectively in education.

This report is commended to all those who seek to learn about *Computers in Education.*

References

1. "Forecast of Canadian Computer Based Services of the Seventies", Bell Canada Headquarters Business Planning, quoted in *Communications, Computers, and Canada.* (Ottawa: Trans-Canada Telephone System, revised Dec. 1971), pp. 55-56.
2. P.C. Brillinger, B.L. Ehle, J.W. Graham, *An Introduction to the Spectre Computer.* (Waterloo: Department of Applied Analysis and Computer Science, University of Waterloo, 1969).
3. John G. Kemeny, *Man and the Computer.* (New York: Charles Scribner's Sons, 1972), pp. 21-43.
4. Adam Osborne, "The Potential Use of Mini-computers in Education", *Journal of Educational Data Processing.* Vol. 8, No. 6, (1971), p. 1.
5. Hobart Loomis and Joseph E. Sucher, "Finding Substitutes in a Hurry", *School Management,* Vol. 16, No. 10, (Oct. 1972), pp. 24-25.
6. "APL — a language for all purposes", *Data Processor,* Vol. 1, No. 2, (1971), pp. 20-21; Canada, Department of Communications, *Branching Out.* Report of the Canadian Computer/Communications Task Force, Vol. 2, (Ottawa: Information Canada, 1972), pp. 105-106.
7. "Service Centre for Schools", *Data Processor,* Vol. 2, No. 4, (1972), pp. 3-6.
8. "The Ties Network", Saturday Review of Education, Vol. 1, No. 4, (April 14, 1972) p. 73; also personal communication from Thomas C. Campbell, Executive Director, Minnesota School Districts Data Processing Joint Board.
9. "At Region IV — Central Computer Facility Serves 700 Schools," *Application Report,* Product Publicity Department, Control Data Corporation, Minneapolis, Minnesota, 1973.
10. *Data Processor,* Vol. 2, No. 4, (1972), pp. 3-6.
11. Robin C. Smith, "AEDS Large School System Survey: a summary", *AEDS Monitor,* Vol. 11, No. 5, (Dec. 1972) p. 5; for complete details see ERIC document number ED 073 679.
12. *Computer Assisted Mathematics Program,* (Glenview, Illinois, Scott Foresman, 1969).
13. Ludwig Braun, *Digital Simulation in Education,* (New York: Polytechnic Institute of Brooklyn, November, 1971), pp. 3-5.
14. *Relevant Educational Applications of Computer Technology,* Course I. Computers in Education: A Survey, Book 7: Computers in Instruction. (San Carlos, California: Technica Education Corp., 1971), pp. 20-26.
15. Listed in: *L.A.C.E. Catalogue of Secondary Education Programs.* (La Crosse, Wisconsin, University of Wisconsin, 1972), p. 18.
16. *Ibid.* p. 47.

17. Allen L. Hammond, "Computer-Assisted Instructions: Two Major Demonstrations", *Science*, Vol. 176, No. 4039, (June 9, 1972), p. 1111.

18. *Huntington II Simulation Program — POLUT.* Teacher's Guide. (Maynard, Mass.: Digital Equipment Corp., 1971) Quote on p. 3.

19. *Branching Out*, p. 111.

20. R.E.A.C.T., *op. cit.* p. 30-31.

21. Patrick Suppes and M. Morningstar, "Computer-assisted instruction", *Science*, Vol. 166, (1969), pp. 343-350.

22. *The Use of Computers in Secondary School Mathematics*, ed. by Dudley Post, (Newburyport, Massachusetts: Entelek 1970), p. 124.

23. Donald L. Bitzer, and others, *Computer-Based Science Education*, (Urbana, Illinois: Computer-based Education Research Laboratory, University of Illinois, 1972), p. 33.

24. *Ibid.* p. 31.

25. Gillian Arsenault, *MATHOP — A computer-assisted instruction course for General Chemistry*, (Burnaby: CAI Centre, Simon Fraser University, 1973), p. 18.

26. Joseph Weizenbaum, "Eliza — a Computer Program for the Study of Natural Language Communication Between Man and Machine", *Communications of the Association for Computing Machinery*, January 1966.

27. R.E.A.C.T., *op. cit.*, p. 32.

28. *Branching Out*, p. 106.

29. Hammond, *op. cit.*, pp. 1110-1112; Bitzer and others, *op. cit.*; *Branching Out*, p. 129.

30. Kenneth J. Stetten, *Toward a Market Success for CAI; an Overview of the TICCIT Program.* (McLean, Virginia: Mitre Corp., 1972); Hammond, *op. cit.*

31. *Branching Out*, p. 107.

32. Mary Stager and Jane Hill, "Computer-assisted instruction: an OISE program in mathematics", *Orbit*, Vol. 4, No. 2, (April, 1973), pp. 20-23.

33. *Branching Out*, p. 107.

34. *Ibid.* p. 107.

35. Post, *op. cit.*, p. 125.

36. William W. Cooley and Robert Glaser, "The Computer and Individualized Instruction", *Science*, (October 31, 1969), p. 574.

37. John M. Finch, "An Overview of Computer-Managed Instruction", *Educational Technology*, Vol. 12, No. 7, (July, 1972), pp. 46-47.

38. M. Westrom, *TAIM The Teacher-Authored-Instruction Manager — a Description*. (Edmonton: Division of Educational Research Services, University of Alberta, 1972).

39. For a discussion of CVIS, PLAN, and ECES see: Jo-Ann Harris, *Tested Practices: Computer Assisted Guidance Systems*. (National Vocational Guidance Association, 1972).

40. This description of SGIS is based on an unpublished report by Reddy Konda, Student Personnel Services, Department of Education.

41. Harris, *op. cit.*, pp. 8-9.

Bibliography

The items in this bibliography have been specially selected for those interested in more information on the use of computers in education. The following codes indicate the location of each item:

EDUC	Education Library, University of Manitoba.
ENG	Engineering Library, University of Manitoba.
SCI	Science Library, University of Manitoba.
CSB	Computer Services Branch, Department of Education, 103 Water Avenue, Winnipeg, Manitoba.
RFB	Department of Education Library, 1181 Portage Avenue, Winnipeg, Manitoba.
WPL	Winnipeg Public Library, 380 William Avenue, Winnipeg, Manitoba.

The bibliographies attached to the course outlines for Computer Science 205 and 305 and Data Processing 202 and 302 can also be consulted.

In addition a file of resource materials collected while researching this report is available at the Computer Services Branch, 103 Water Avenue.

ERIC Documents

The Educational Resources Information Centre (ERIC) operates a clearinghouse for documents relating to education. These documents are available in microfiche form from the Education Library, University of Manitoba or from the Department of Education Library, 1181 Portage Avenue.

The ERIC Document Accession Number is listed with each document in this bibliography. Please quote it when asking the librarian for assistance.

I General

Books:

Bernstein, Jeremy, *The Analytical Engine: Computers, Past, Present, and Future.* Random House, New York, 1964. (Good for historical information only.) SCI, WPL

Canada, Department of Communications, *Branching Out*, Report of the Canadian Computer/Communications Task Force. Vol. 2, Ottawa, Information Canada, 1972. WPL

Computers. . .A Beginning. A Continuing Education Book, Corvallis, Oregon, 1972. EDUC

Computers in Education. Their Use and Cost. Education Automation Monograph. Part 1: President's Science Advisory Committee, Part 2: Michael G. Sovereign, American Data Processing Inc., Detroit, N.D. RFB

Crowley, Thomas H., *Understanding Computers.* McGraw Hill, New York, 1967. CSB

Fry, T.F., *Computer Appreciation.* Philosophical Library, New York, 1971. ENG

Hawkes, Nigel, *The Computer Revolution.* Thames and Hudson, London, 1971. CSB, RFB, WPL

Information Processing in Society. McLean-Hunter Learning Materials Co., Toronto, 1972. CSB

Johnson, M. Clemens, *Educational Uses of the Computer: An Introduction.* Rand McNally, Chicago, 1971. EDUC

Kemeny, John G., *Man and the Computer.* Charles Scribner's Sons, New York, 1972. RFB, WPL

Layman's Guide to the Use of Computers. Association for Educational Data Systems, Washington, D.C., 1971. RFB

Martin, James and Adrian R.D. Norman, *The Computerized Society.* Prentice-Hall, Englewood Cliffs, N.J., 1970. SCI, WPL

Nikolaieff, George, A., ed., *Computers and Society.* H.W. Wilson, New York, 1970. SCI, WPL

Pylyshyn, Zenon W., ed., *Perspectives on the Computer Revolution.* Prentice-Hall, Englewood Cliffs, N.J., 1970. ENG

Relevant Educational Applications of Computer Technology. Course I for Teachers/Administrators (Books 1 — 10), Course II for Teachers (Books 1 — 8), Course III for Administrators, (Books 1 — 10). Technica Education Corp., San Carlos, California, 1971. CSB

Squire, Enid, *The Computer: An Everyday Machine.* Addison-Wesley, Toronto, 1972. CSB

Articles:

Osborne, Adam, "The Potential Use of Mini-Computers in Education." Journal of Educational Data Processing. Vol. 8, No. 6, p. 1-10, 1971. EDUC

II Administrative Uses

Books:

Kenney, James B. and R. Robert Rentz, *Automation and Control of Public School Instructional Records.* F.E. Peacock, Itasca, Illinois, 1970. EDUC

Knezevich, Stephen J. and Glen G. Eye, eds., *Instructional Technology and the School Administrator.* American Association of School Administrators, Washington, 1970. EDUC

Principles of Automatic Data Processing. Data Processing Management Association, Park Ridge, Illinois, 1965. CSB, RFB

Stoker, Howard W., *Automated Data Processing in Testing.* Houghton Mifflin, Boston, 1968. EDUC

Articles:

"Building Timetables with a Touch of Serendipity", *School Progress.* Vol. 41, No. 7, pp. 18-19, July, 1972. RFB, EDUC

Loomis, Hobart and Joseph E. Sucher, "Finding Substitutes in a Hurry", *School Management,* Vol. 16, No. 10, p. 24-25, October, 1972.

Schoonover, Kenneth P., and others, "District Budget on a Computer", *School Management.* Vol. 15, No. 12, pp. 16-19, December, 1971. EDUC, RFB

ERIC Documents:

Farquhar, John, *An Information System for Educational Management. Executive Summary.* Rand Corporation, Santa Monica, Ca. 30 p., March 1972. (Los Angeles Unified School District). ED 067 730

Floyd, Jerald D., *The Computer: An Administrative Dilemma.* Northern Illinois University, De Kalb, 13 p., 1972. ED 069 116

Piele, Philip K., *Computer Applications in Class and Transportation Scheduling, Educational Management Review Series No. 1,* University of Oregon, Eugene, 8 p., October, 1971. ED 057 436

Smith, Robin C., *The AEDS Large School System Survey. Report of Findings.* Association for Educational Data Systems, Washington, 57 p., November 1972. ED 073 679

Witkin, Belle Ruth, *Management Information Systems: Applications to Educational Administration.* Alameda County School District, Hayward, California, 146 p., 1971. ED 057 608

III Instructional

Books:

Computer Assisted Mathematics Program. Scott Foresman, Glenview, Illinois, 1969. CSB, EDUC

Hicks, B.L., and S. Hunka, *The Teacher and the Computer.* Saunders, Philadelphia, 1972. EDUC

Koch, Warren J., *The Use of Computers in Instruction in Secondary Schools.* National Association of Secondary School Principals, 1972. CSB

Margolin, Joseph B., and Marion R. Misch, *Computers in the Classroom: An Interdisciplinary View of Trends and Alternatives.* Spartan Books, New York, 1970. EDUC

Post, Dudley L., *The Use of Computers in Secondary School Mathematics.* Entelek, Newburyport, Mass., 1970. EDUC

Articles:

Barnard, Charles N., "Counselling by Computer" *Education Digest.* Vol. 38, No. 1, pp. 19-22, September, 1972. RFB, EDUC

Cooley, William W., and Robert Glaser, "The Computer and Individualized Instruction". *Science.* Oct. 31, 1969, p. 574-582. SCI, ENG

Finch, John M., "Computer-Managed Instruction: An Annotated Bibliography". *Audiovisual Instruction.* Vol. 17, No. 3, pp. 72-74, 76, March 1972. RFB, EDUC

Finch, John M., "An Overview of Computer-Managed Instruction". *Education Technology.* Vol. XII, No. 7, pp. 46-47, July, 1972. RFB, EDUC

Fromer, Robert, "Distinctions Between CAI and CMI Systems". *Educational Technology.* Vol. XII, No. 5, pp. 30-31, May, 1972. EDUC, RFB

Gibb, E. Glenadine, "The Computer — A Facilitator in Management and Instruction". *Mathematics Teacher.* Vol. 66, No. 1, p. 6, January, 1973. EDUC, RFB

Hammond, Allen L., "Computer-Assisted Instruction: Many Efforts, Mixed Results". *Science.* Vol. 176, No. 4038, pp. 1005-1006, June 2, 1972. SCI, ENG

Hammond, Allen L., "Computer-Assisted Instruction: Two Major Demonstrations". *Science.* Vol. 176, No. 4039, pp. 1110-1112, June 9, 1972. SCI, ENG

Jerman, Max, "Use of Computers to Individualize Instruction". *Mathematics Teacher.* Vol. 65, No. 5, pp. 395, 466-471, May, 1972. EDUC, RFB

Koch, Warren J., "Using Time Shared Computers in Secondary Schools: A Status Report". *NASSP Bulletin.* Vol. 56, No. 363, pp. 46-54, April, 1972. RFB, EDUC

Lamon, William, "The Computer: An Instructional Aid in the Secondary Mathematics Classroom". *Journal of Educational Data Processing.* Vol. 8, No. 6, pp. 11-21, 1971. RFB, EDUC

Leonard, W. Patrick, "Acquiring a TSS". *Audiovisual Instruction.* Vol. 17, No. 4, pp. 24-26, April, 1972. EDUC, RFB

Molenda, Michael H., "How a School System Gets into CAI". *Audiovisual Instruction.* Vol. 16, No. 6, pp. 86-87, June/July, 1971. EDUC, RFB

Myers, Sheldon S., "A Study of the Educational Technologies of Computer-Assisted Instruction, Instructional Television, and Classroom Films, Based on a Tour of Sites". *Educational Product Report Number 45.* Vol. 5, No. 9, pp. 6-47, June, 1972. EDUC

Resta, Paul E., Joel E. Strandberg, and Edward Hirsch, "Instructional Management Systems Using Computers". *Audiovisual Instruction.* Vol. 16, No. 10, pp. 28-31, December, 1971. EDUC, RFB

Stocker, H. Robert, "Computer Assisted Instruction". *Delta Pi Epsilon Journal.* Vol. 14, No. 3, pp. 15-24, May, 1972. EDUC

ERIC Documents:

Anastasio, Ernest J., and Judith S. Morgan, *Factors Inhibiting the Use of Computers in Instruction.* EDUCOM, Interuniversity Communications Council, Princeton, N.J., 130 p., 1972. ED 069 219

Blackwell, F.W., *The Probable State of Computer Technology by 1980,* With Some Implications for Education. Sept. 1971. ED 057 605

Block, Karen K., *Strategies in Computer Assisted Instruction: A Selective Overview.* Learning Research and Development Center, University of Pittsburgh, Pa., 1970. ED 049 614

Braun, Ludwig, *Digital Simulation in Education.* Brooklyn Polytechnic Institute, New York, 27 p., November, 1971. ED 061 744

Computer-Assisted Instruction: A general Discussion and Case Study. Bureau of Training, Civil Service Commission, Washington, D.C., 23 p., August, 1971. ED 054 621

Dunn, Alex and Jean Wastler, *Computer-Assisted Instruction Project. Project REFLECT. Final Report.* Montgomery County Public Schools, Rockville, Md. 502 p., 1972. (a 3 year CAI project in a suburb of Washington, D.C.) ED 066 876

Dwyer, Thomas A., *Some Principles for the Human Use of Computers in Education.* Department of Computer Science, University of Pittsburgh, Penn., 1970. ED 053 566

Fritz, Kentner V. and Lynn B. Levy, *Introduction to Computer Managed Instruction and the Automated Instructional Management System.* Counselling Center, Wisconsin University, Madison, Wisconsin, 26 p., June, 1972. ED 069 757

Hansen, Duncan N. and Barbara Johnson, *CAI Myths that Need to be Destroyed and CAI Myths that We Ought to Create.* Computer-Assisted Instruction Center, Florida State University, Tallahassee, 31 p., June, 1971. ED 054 650

Hansen, Duncan N., *The Role of Computers in Education during the 70's.* Computer-Assisted Instruction Center, Florida State University, Tallahassee, May 1970. ED 043 238

Lyman, Elizabeth R., *A Summary of Plato Curriculum and Research Materials.* Computer-Based Education Research Laboratory, University of Illinois, Urbana, 52 p., August, 1972. ED 066 931

Recommendations Regarding Computers in High School Education. Conference Board of the Mathematical Sciences, Washington, 36 p., April, 1972. ED 064 136

Sass, Richard E., *A Computer-Based Instructional Management Program for Classroom Use.* Learning Research and Development Center, University of Pittsburgh, Pennsylvania, 76 p., May, 1971. ED 052 621

Singh, Jai P. and Robert P. Morgan, *Computer-Based Instruction: A Background Paper on its Status, Cost/Effectiveness, and Telecommunications Requirements.* Washington University, St. Louis, Missouri, 40 p., April, 1971. ED 055 429

Stetten, Kenneth J., *Toward a Market Success for CAI: An Overview of the TICCIT Program.* Mitre Corporation, McLean, Virginia, 66 p., June, 1972. ED 066 034

Suppes, Patrick and others, *Teacher's Handbook for CAI Courses.* Institute for Mathematical Studies in Social Science, Stanford University, California, 141 p., September, 1971. ED 054 620

Wightman, Lawrence, *What Computers Can Do Now*, School of Education, University of Massachusetts, Amherst, 13 p., 1970. ED 042 205

Glossary

access time — the time needed to locate and retrieve data from a computer memory device.

accumulator — a special location in the arithmetic/logic unit used for arithmetic operations.

address — a unique label or number used to denote a particular word in a computer's memory.

algorithm — a step-by-step procedure for finding the solution to a problem.

APL — A Programming Language — an interactive programming language used for problem solving — developed by IBM Corp.

arithmetic/logic unit — the part of a computer which performs arithmetic and logical operations.

assembler — a program that translates assembly language programs into machine language.

assembly language — a low level computer language which permits machine instructions to be written in symbolic form.

audio response unit — an output device which constructs vocal responses from a pre-recorded vocabulary.

authoring language — a special language used for writing computer assisted instruction programs.

automatic data processing — the use of electronic or mechanical devices to process data.

auxiliary storage — storage which is used in addition to the main storage of a computer e.g. magnetic tape, disk, or drum.

BASIC — Beginner's All-Purpose Symbolic Instruction Code — an easily-learned interactive programming language which uses simple English commands. It was developed at Dartmouth College.

batch processing — a technique in which a group of programs to be run are submitted together and processed automatically one after another by the computer.

binary digit — (bit) — either a 0 or a 1, a digit in the binary system. All information inside the computer is represented in terms of bits.

bug — a mistake in a computer program; removed by "de-bugging".

card reader — an input device which senses coded information on cards and stores it in the computer.

card punch — an output device which stores information on cards by punching holes in them.

cathode ray tube (CRT) — a television-like screen or display often used as an input/output device for a computer.

central processing unit (CPU) — the portion of the computer that interprets and carries out the instructions in the computer's memory. It consists of the arithmetic/logic unit, control unit, and main storage unit.

COBOL — COmmon Business-Oriented Language — a computer language intended for business use.

compiler — a program which translates programs written in a high level language into a machine level language.

computer — an electronic device which processes information under the control of a stored program.

computer-assisted instruction — method of using a computer system to present individualized instructional material.

computer literacy — a course whose aim is to provide all students with a general knowledge of the computer and its role in their lives.

computer managed instruction — using the computer to assist the teacher in the management of individualized instruction.

computer science — the study of computers and their languages.

control unit — the portion of the central processing unit which interprets and executes the instructions in proper sequence and controls the function of all other units in accordance with the instructions.

core storage — the most common type of main storage. It represents information by means of tiny doughnut-shaped cores of magnetic material.

data preparation — the process of converting information from human-readable form to machine-readable form.

data processing —

1. the manipulation of data according to precise rules of procedure to produce information.
2. the study of the use of computers in business.

debugging — the process of locating and correcting mistakes in a computer program.

dialogue system — the most sophisticated type of computer-assisted instruction. It permits the student and the computer to carry on an extensive dialogue about the subject matter.

direct access — a storage method in which access to the next position from which information is to be obtained is in no way dependent on the position from which information was previously obtained.

documentation — a set of notes accompanying a computer program which describe its purpose, details of the algorithm used, samples of the input and output, and complete instructions on the use of the program.

drill and practice system — least complex form of computer-assisted instruction. It is used to drill and review students on previously taught material.

electronic data processing — EDP — data processing carried out by a computer, especially in the area of business.

execution phase — second phase in running a program, during which the machine language instructions are executed one at a time by the computer.

flowchart — graphical representation of the steps used to solve a problem. The preparation of a flowchart is usually the first step in writing a computer program.

FORTRAN — FORmula TRANslation — a programming language designed primarily for scientific and mathematical uses.

hardware — the physical equipment and electronic circuitry making up a computer.

high level language — a computer language with a well defined grammar, vocabulary, and punctuation which is convenient to use in writing programs. It must first be translated into machine language by a program called a compiler before it can be executed by the computer. Each statement in the high level language is expanded into several machine level instructions.

input — transfer of information from an external medium such as paper, cards, or tape into the main storage of the computer.

integrated data base — the complete set of computerized files used by a management information system. Cross references are established between all these files to minimize duplication and facilitate retrieval of information.

keypunch — a machine which encodes information on cards by punching holes in them.

light pen — a photo-electric pen used for pointing at or drawing on a CRT display.

line printer — device which prints computer output on standard computer paper.

low level language — either a machine language or assembly language.

machine language — set of numeric instructions which can be directly interpreted by the circuitry of the computer.

magnetic disk — a disk coated with magnetic material upon which information may be stored as a series of magnetized spots; can use direct or sequential access.

magnetic drum — a cylinder coated with magnetic material upon which information may be stored as a series of magnetized spots; can use direct or sequential access.

magnetic tape — a tape coated with magnetic material upon which information may be stored as a series of magnetized spots; uses sequential access only.

main storage — the part of the central processing unit where program instructions and data are stored.

management information system — a computerized system for providing management with adequate, accurate, and timely information for decision making.

master file — a file of information which changes relatively infrequently.

memory — see storage.

microsecond — one millionth of a second.

millisecond — one thousandth of a second.

mnemonic — an abbreviation or letter combination that is easy to remember, used in assembly language programming.

nanosecond — one billionth of a second.

object program — the machine language program produced by a compiler from the original source program in a high level language.

off-line — not under the control of the central processing unit.

OMR card — (optical mark reader card) — a special type of card upon which information can be encoded as pencil marks instead of punched holes. It can be read by a modified card reader.

on-line — under the direct control of the central processing unit.

operand — the part of a machine language instruction which specifies a location in main storage to be used in the operation.

operation code — the part of a machine language instruction specifying which operation is to be performed.

output — transfer of information from the main storage of the computer to an external medium such as paper or a CRT display.

PL/1 — Programming Language One — a programming language suited to both scientific and business applications.

program — a sequence of instructions which directs a computer to solve a particular type of problem.

random access — see direct access.

read — to accept or copy information from an input device into a computer.

read/write head — a small electromagnetic unit used for reading, recording, or erasing magnetized spots on a magnetic tape, disk, or drum.

record — a group of related data items.

remote device — an input/output unit or other piece of equipment which is removed from the computer centre but connected by a communication line.

sequential access — a storage method in which the stored items of information become available only in a one after the other sequence whether or not all the information or only some of it is desired.

simulation — the use of a mathematical model to explore the characteristics of a real-world phenomenon.

software — the collection of programs such as compilers, assemblers, etc., supplied by the manufacturer to make the computer function effectively; as opposed to hardware.

source program — a program written in other than machine language which must be translated by the computer into machine language before use.

storage — the part of the computer into which information can be copied, which will hold this information, and from which the information can be obtained for use at a later time.

terminal — a device by which data may enter or leave the computer system, eg. typewriter terminal, CRT terminal.

time sharing — a technique by which a computer can service many users simultaneously. The users' terminals can be located at any distance from the computer.

time slice — the small amount of time given each user's program in turn in a time-sharing system.

transaction file — transactions accumulated as a batch ready for processing against the master file.

translation phase — first phase of running a program during which the statements in the high level language are translated into machine language.

translator — a program which translates instructions from the programming language in which they are written into a machine level language which can be interpreted directly by the computer. Two classes of translators are assemblers and compilers.

tutorial system — the type of computer-assisted instruction where the computer assumes primary responsibility for teaching new material.

word — a set of bits which occupies one storage location and is manipulated as a unit by the computer.

write — to copy information from a computer to an output device.