DOCUMENT RESUME

ED 041 455	88	EM 008 119

AUTHOR	Edwards, Judith B.
TITLE	Computer Instruction: Planning and Practice.
INSTITUTION	Northwest Regional Educational Lab., Portland, Oreg.
SPONS AGENCY	Department of Health, Education, and Welfare, Washington, D.C. National Center for Educational Research and Development.
PUB DATE	Sep 69
NOTE	133p.
AVAILABLE FROM	Northwest Regional Educational Laboratory, 400 Lindsay Building, 710 S.W. Second Avenue, Portland, Oregon 97204
EDRS PRICE	EDRS Price MF-$0.50 HC-$6.75
DESCRIPTORS	*Computer Science Education, Course Descriptions, *Curriculum Guides, High School Curriculum, Technical Education

ABSTRACT

On the basis of the experience gained in a Title III project called Computer Instruction NETWORK, the author presents some guidelines for the planning and implementation of a course at the high school level dealing with computers. The guide lists the first steps to be taken in setting up such a program, describes a teacher training program, and provides help in selecting a programing language. Choosing a programing language suitable for special purpose use, such as business education or mathematical problem solving, is discussed. The factors involved in selecting computers, terminals, input equipment, a time sharing scheme, and communications devices are noted. The cost of various equipment and training devices necessary for such a program is estimated. An annotated bibliography is appended. (JY)

Computer Instruction: Planning and Practice

Judith B. Edwards

Northwest Regional Educational Laboratory

400 Lindsay Building • 710 S.W. Second Avenue • Portland, Oregon 97204

Computer Instruction: Planning and Practice

Judith B. Edwards

Northwest Regional Educational Laboratory

400 Lindsay Building • 710 S.W. Second Avenue • Portland, Oregon 97204

FOREWORD

Educators now have available an exciting new instructional "tool"--The Computer.

One outstanding example of how the computer can be used to improve education is demonstrated by the Computer Instruction Network, a Title III project.

It is a strong conviction of the Northwest Regional Educational Laboratory that information about this project should be widely disseminated so that other educators may benefit from the experiences of students, teachers and administrators at schools in the Computer Instruction Network.

Lawrence D. Fish
Executive Director

Portland, 1969

September 1969

Published by the Northwest Regional Educational Laboratory, a private nonprofit corporation supported in part as a regional educational laboratory by funds from the United States Office of Education, Department of Health, Education and Welfare. The opinions expressed in this publication do not necessarily reflect the position or policy of the Office of Education, and no official endorsement by the Office of Education should be inferred.

Northwest Regional Educational Laboratory, 400 Lindsay Building, 710 S.W. Second Avenue, Portland, Oregon 97204

PREFACE

In 1962 two teachers in Oregon schools 30 miles apart began experimenting with teaching about computers. During the next three years, Mrs. Marian Putnam at North Salem High School and Mr. Bud Pembroke at Lake Oswego High School successfully incorporated one-month units in computer concepts into existing mathematics courses.

Students in these classes learned about computers by actually programming and operating a simple digital computer. The classroom computer had been developed several years earlier at Oregon College of Education with a grant from the National Science Foundation.

The Salem School District and the State Department of Education, together with the Marion County Intermediate Education District, decided in 1965 the good things happening in these two schools should be studied and expanded to other districts.

A proposal for a planning grant was written and funded under Title III of the Elementary and Secondary Education Act. Forty secondary schools in a four-county area participated in the eight-month planning phase which began in the summer of 1966. It resulted in an operational grant for a program, which began in the summer of 1967. The Title III project, called Computer Instruction NETWORK, has operated through the Marion County IED and has received some $100,000 per year for each of three years.

Several new procedures for computer instruction have been tried and evaluated. The results of these experiences are reported in this monograph to help other schools plan and implement programs for teaching with and about computers.

CONTENTS

CONTENTS

COMPUTER INSTRUCTION: A RATIONALE

Introducing the special September 1966 issue of The Saturday Review devoted to "The Computerized Age," the editors said, "Few technological developments are formidable enough to mark turning points in human history. Two such phenomena have occurred in our time: The atomic bomb and the computer.... The implications of the computer as yet are only faintly comprehended."

The computer is changing human history; no longer is this debatable. As a tool for extending man's thinking powers, the computer portends something as consequential as the industrial age when power was applied to supplement man's muscles. The age of the computer has developed so quickly and with such an impact that the man in the street views the computer with awe, fear and mistrust. The information available to students and their parents in newspapers, magazines, television and films usually emphasizes the dramatic nature of computers and automation and plays down the role of man. Consequently, the Man-Machine System is viewed as a Machine-Man System. The most dangerous of the misconceptions about the computer is its potential for "taking over" or controlling man and his world. The possibility of the myth becoming reality increases if educators ignore the need to prepare students for the automated age.

Students entering high school in 1970 will be 45 years old in the year 2000. By that time, two significant realities of the computerized age will be a part of their everyday life: the computer utility and the cybernated system.

Widespread use of the computer utility will mean a computer terminal in every home, as familiar as the telephone or cable television. Telephones and television, however, do not have the vast implications for social and cultural change which computers have. The cashless, checkless society is technically feasible now and sure to be a reality in the year 2000. Banking and all exchange of goods can be accomplished using computers without the transfer of actual money.

The implementation of these computer based systems is not without problems, however. "Total information" at everyone's fingertips raises questions of personal privacy which are only beginning to be examined, for example.

"Unemployment resulting from automation would be greater right now except that industries are holding back--at a sacrifice to their projects," says Richard Bellman of the RAND Corporation. "The scientific know-how to automate U.S. industry completely is already available and is certain to be used." When cybernated systems do, indeed, completely automate industry, a new ethic other than the sanctity of work will have to be found. Meanwhile, the computer--the cause of all this--remains little understood and often poorly used.

Two educational tasks are vital in the changes being wrought by the computer. Education must begin these tasks if students are to receive a relevant education. The first of these, which has become the primary goal of the Computer Instruction NETWORK, is to provide a degree of "computer appreciation" for as wide a segment as possible. The second--more difficult to tackle and with much broader implications--is to prepare students to live in a world so drastically changed by automation that the old values, occupations and roles no longer have meaning.

The "instructional" roles for the computer, as opposed to data processing for school administration, include prevocational training, problem solving as a "curriculum extender," and computer appreciation units or courses.

Computer appreciation is more than the study of automation. It includes understanding the simplicity of the machine responsible for the computerized age. This basic understanding of "what is a computer?" should be available to all students. It should not be considered a technical topic worthy of examination only in mathematics or vocational courses. The computer, itself, need not be studied in detail or at a high level of complexity, but study of a simple computer with a simple language--even a hypothetical one--can remove the mystery. The notion computers do not "think," but must be instructed by a human being, is not easy for a student to understand until he tries instructing a computer himself.

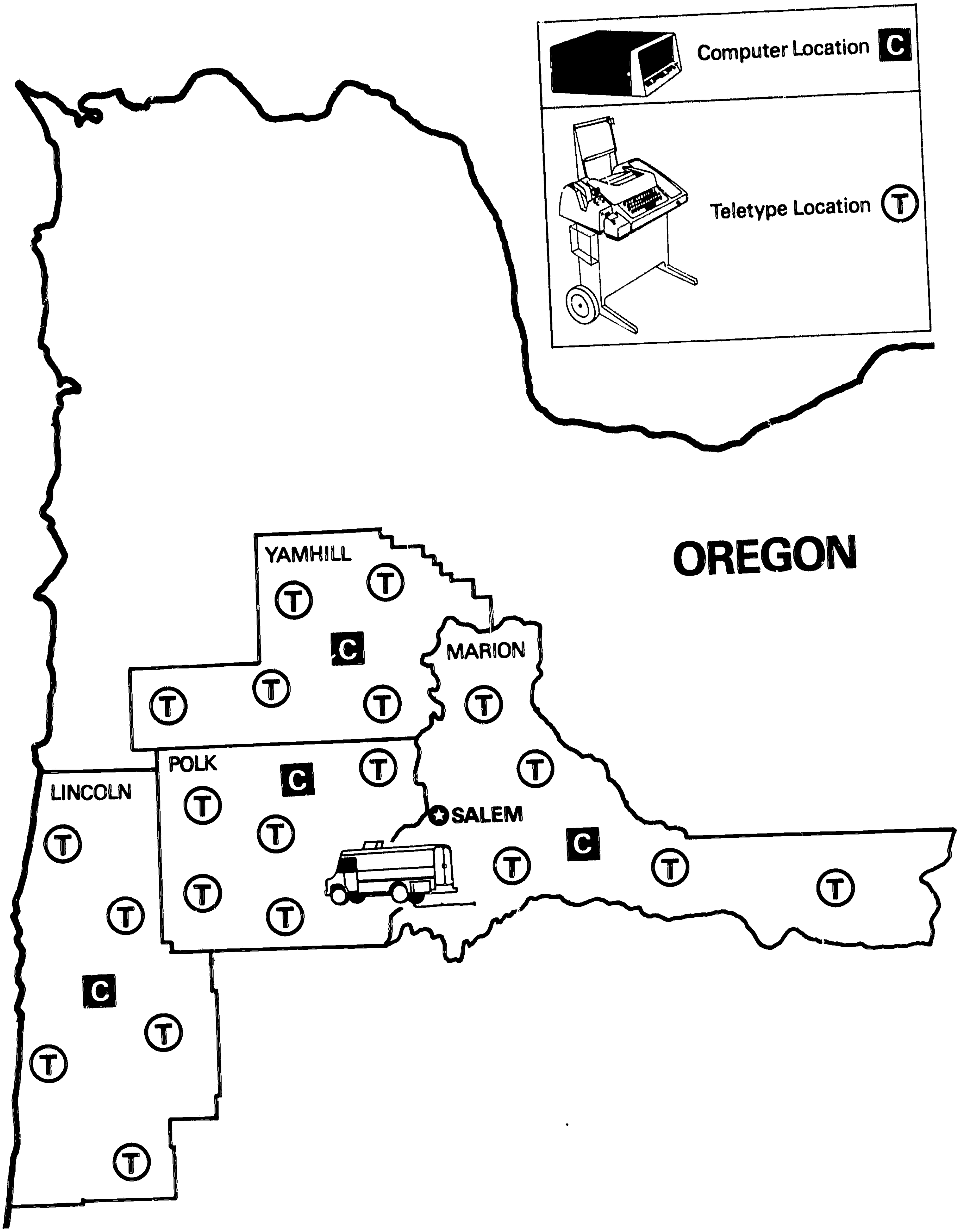
Awareness of the potential power of the computer is as important as the study of its simplicity. Developing this understanding is as vital for the student who will be affected only indirectly by computers in his lifetime as for the college bound engineering student. Only the informed, imaginative and thoughtful use of computers by the consumer can prevent the age of computerization from being a threat to mankind and ultimately a human tragedy.

The preNETWORK pilot classes demonstrated that student operation of an uncomplicated computer is the most effective way to teach about computers. The complexities of machine language need not be a major area of study. With a little practice using a simple machine language, students comprehend the concept of computer memory, how a computer is programmed to solve problems, the movement of information into and inside the machine, and the necessity for the computer operator to use careful and logical problem analysis techniques. A bit of serendipity apparently motivates students--particularly underachievers--when they are allowed to operate the machine themselves.

The Computer Instruction NETWORK, then, was planned to develop both students' and teachers' computer appreciation--their understanding of the computer itself and an appreciation of its potential for good or evil. Secondly, students who achieved some computer appreciation were given the opportunity to use the computer itself for developing problem solving skills.

The particular constraints the NETWORK faced included the very high percentage of small rural schools in the project area (37 out of 40), a lack of teachers with even slight knowledge of computers, a need for classroom handson experience in each school and a limited budget. Most schools operate with these same constraints. The schools in the NETWORK region had been reluctant to venture into this area on their own because of the multiplicity of decisions to be made and the lack of funds, qualified teachers and appropriate teacher-training courses, continuing professional assistance and outside impetus. The many decisions to be made probably is the most dismaying prospect of all. Crucial decisions must be made about curriculum, emphasis, equipment, software, materials, methods and teacher training. Few are qualified, or feel qualified, to make such decisions.

The NETWORK provided the outside impetus, the continuing professional assistance, teacher training and funds for schools to begin. Small computers were rotated among schools to provide handson instruction, and Teletype terminals were installed at the schools to provide for remote problem solving and communication with the NETWORK office and other schools. One larger computer installed permanently in a large van travelled among rural schools. Teachers were trained by NETWORK staff in a continuing series of formal courses and short workshops, and the nearby colleges and universities began to provide appropriate courses for teachers. Curriculum planning and development of instructional materials continues.


Computer Location C
Teletype Location T
OREGON
YAMHILL
MARION
POLK
LINCOLN
SALEM

THE PROPER ORDER

Many schools start computer instruction because of the persistence of a computer salesman or the availability--somewhere--of a computer. The equipment, then, dictates the objectives, methods and sometimes even the curriculum.

This is decidedly cart first. Equipment selection should come last, after all other important decisions have been made. Equipment should be selected on the basis of these decisions. A careful study involving administrators and curriculum coordinators, as well as teachers from several subject areas, should result in decisions in the following order:

Curricular areas to be involved

Teacher training

Unit and course content

Selection and/or adaptation of programming languages to to be taught and used

Selection and development of resource materials

Equipment

The first, curricular areas, often is made "easy" when a mathematics teacher indicates interest in initiating a computer program. Too often, however, this means decisions are made on the basis of "what is needed for mathematics" rather than "what is needed for computer instruction." If mathematics teachers make all decisions in the early stages, business teachers give up in disgust, social studies teachers turn their backs and science teachers may or may not be slightly interested. It is altogether possible to have what is needed for both mathematics and computer instruction, if the early planning is done by a cross-curricular group of teachers and administrators, perhaps with the help of competent professionals. If teacher training can begin a year or two in advance, planning is made immensely easier.

The specification of course and unit content implies either the existence of trained teachers or the assistance of competent outside help. The opinions of several outside consultants, however, should be sought rather than relying on the possible prejudices of one individual.

The selection of programming languages is critical. Too often, FORTRAN is used for instruction simply because it is available. However, for nonvocational training or problem solving, the simplest possible language to teach and use is the best choice. An example is the conversational BASIC. For simple handson practice with a computer, the most straightforward and simple machine language available is preferred. If one does not have freedom of choice, appropriate subsets of the available languages may be identified for instructional use.

Resource materials are more likely to be "developed" than "selected." Seldom is there a single text available which is exactly appropriate for one's objectives. Films often are outdated and poorly done. And manufacturer's reference manuals are exactly that--meant to be used for reference by a well trained computer user, not for instructional purposes.

Finally, equipment selection can follow evaluation of all computers and terminals. If the school district is getting a computer for administrative purposes, certainly the specifications of the computer instruction planning committee should be given as high a priority as the administrative considerations. If no effective compromise can be reached, equipment for computer instruction should be planned for in addition to the district's administrative system. It is an unfortunate fact of life in education that data processing and administrative needs of school systems usually are met with alacrity and money, while computer instruction must "wait until next year."

PLANNING FOR COMPUTER INSTRUCTION

A maximum of six people should form a committee to begin the planning for computer instruction. This planning committee can include one representative from each of four curricular areas: social studies, science, business education and mathematics. In addition, a district administrator or curriculum coordinator should be involved. If the committee gets much larger, little effective action will be taken.

In planning the committee must realize any unit or course will demand more time of the teacher than any other course he has ever taught. A lecture-type approach usually is discarded after the first few days in favor of a lab-tutorial system. Since resource materials usually are not appropriate for what is being taught, teachers often must develop their own audiovisual aids and mimeographed materials. Released time for such activities, and help in supervising computer labs, relieves the pilot teacher of some of the burden.

Once formed, the committee can begin immediately to find answers to the many vital questions.

What computing equipment is available locally?

Investigate local businesses, banks, the school district office, government agencies, colleges and universities. Find out about equipment availability (times as well as student access), costs, languages used, input media required (punched cards, paper tape or remote keyboard), auxiliary equipment required in the school (keypunch or teletype), and personnel available for assistance.

What teachers in the district have had training or experience with computers?

These teachers can serve as leaders in planning curriculum content.

What teachers, though untrained, are interested and willing to teach about or with computers?

Teachers identified should be the first ones trained. An enthusiastic group of trained teachers from several subject areas lessens initial anguish.

What professional advice and assistance is available?

Ask for help or recommendations from professional organizations such as the National Council of Teachers of Mathematics, Association of Business Educators, Association for Educational Data Systems, Data Processing Management Association and Association for Computing Machinery. Inquire at universities, regional educational laboratories, federal projects, the state educational agency, state government, local businesses, business training schools and technical-vocational schools.

Often manufacturers can provide valuable free materials and sample curricula, but keep in mind this source probably will provide you with advice designed to sell a particular computer.

What funds are available?

In addition to funds from the local district budget, ask the state educational agency what funds might be sought from other agencies (including the state). Possibilities include federal grant programs, vocational education programs, foundations and local businesses. In some well-to-do districts, students pay for their own time on computers or terminals.

How can teachers be trained?

If no appropriate courses for teachers are available at nearby universities--and there probably are none--start pressure immediately for such courses. Ask the state educational agency what extension courses are offered, or might be. Some manufacturers offer short courses at no cost. The National Science Foundation sponsors a number of excellent summer institutes and conferences at universities around the country. Ask the state chapter (if there is one) of the Association for Educational Data Systems for information about available computer courses. Or contract with a consultant to come into the district and offer workshops before school starts in the fall.

What are the objectives for computer instruction in each of four curricular areas (math, science, social studies and business education), and what objectives can be defined for cross-curricular needs?

Defining these objectives takes time and thought. The committee should study the objectives developed in other districts or projects. A mathematics teacher might wish to use the computer only as a problem solving aid or as a "mathematics laboratory," but a social studies teacher may wish to examine the social and cultural implications of automation. Prevocational computer training could be the goal of a business education teacher, and a scientist might decide to extend the existing curriculum in physics, chemistry or biology by exploring and experimenting with the aid of the computer. All four teachers can combine talents to teach about the computer itself.

What content should be taught to reach the objectives?

Content designed to meet instructional objectives should be specified with outside professional assistance or by adapting from programs in other schools. Since this will be a pilot program, syllabi, outlines and guides probably will need revision after the first trial and evaluation. Some discoveries made in operation of the NETWORK were: "History of the

Computer" often is a tedious topic consuming time better used on other topics; "problem solving techniques" often is passed over too quickly; business education "data processing" courses sometimes turn out to be keypunching, which is a skill with questionable saleability in tomorrow's world; sophisticated programming techniques and complex languages should not be taught at all, except possibly for vocational purposes. Advice on content, if solicited from computer professionals who are not educators, is likely to be oriented toward vocational training, with emphasis on computer programming and operation skills.

What length unit or course can be implemented best in the beginning?

A short unit of a month or two incorporated in an existing course is the smoothest beginning. The mathematics curriculum is the most obvious vehicle. Business education is another. Social studies, while not so obvious a choice, could certainly incorporate a unit on computers and automation as part of a modern problems class. If the course is an elective, try a team-taught semester course. A full year's course usually tries the patience, skill, inventiveness and knowledge of a novice teacher of computer concepts.

What programming languages should be taught?

The choice of a programming language will depend upon objectives. Criteria for judging and selecting languages appear in the section titled Languages for Computer Instruction.

What equipment or terminals will be needed?

This will depend on the instructional objectives and course content, available funds, languages chosen and number of students to be served. Usually necessary is some additional auxiliary equipment--extra keypunches, Teletypes, optical card readers which will read both punched cards and cards marked with a pencil, or other devices to prepare programs.

What will costs be?

Costs should include computer purchase or lease, maintenance contracts, Teletype lease or purchase, line charges and contract charges for time sharing, long distance charges if time sharing is not on a local computer, and cost of auxiliary equipment. Teacher training and released time for teachers to plan, prepare materials and visit computer instruction programs in other schools also should be included.

How much can a school afford?

If a district must depend on local funds for total support, a modest beginning should be planned. Initially, only funds to support teacher training need be committed. Short units might be taught using "on approval" equipment or borrowed equipment, or by sharing costs with other schools or districts. A successful program probably will always require funds to make additional equipment available for student use. Teacher training is a never ending project. Teachers who have developed a beginning level of knowledge and experience will be the most vociferous in demanding more advanced courses. The district should plan to continue to provide financial aid to these teachers for further training.

TEACHER TRAINING

Although the C. I. NETWORK plan called for an intensive summer of teacher training to precede implementation, budget reductions made it impossible. Instead, the NETWORK was implemented slowly and gradually, first using only teachers with previous training. Four evening courses for teachers were taught the first fall term and two courses each term since. On the basis of this experience, teacher training started well in advance and continued on a regular basis is urged.

Teacher training never ends. Once a nucleus of teachers has a good background in basic computer concepts and programming, a demand is created for advanced courses: programming in various languages, advanced programming techniques, special courses for particular curricular areas and advanced educational technology. At this time, an advanced course in educational technology does not exist, but should include computer assisted instruction languages, practical experience in CAI development and indepth exploration of the ways a computer can expand or extend the curriculum and ease a teacher's load. Meanwhile, attrition and the exposure of other teachers maintains the need for introductory courses.

The mechanics of establishing a course for graduate credit, approved through a university, is sometimes a tedious one, but is worth doing if teachers in the program need credit for their graduate program. Often the credit course is an enticement for teachers who otherwise might not enroll in a computer course. Most districts have an established inservice program and easily can include computer courses. In Oregon, the State System of Higher Education, Division of Continuing Education (DCE) offers credit courses with approved university prefixes on an extension basis. Usually, a one-term, three-quarter-hour credit course offered through DCE will meet for 30 class hours, one night a week for 11 weeks. This format was used by C. I. NETWORK, and it proved a convenient system.

For several years the National Science Foundation (NSF) has sponsored summer institutes and conferences on university campuses to train teachers in computer technology. Most of these institutes are computer mathematics oriented and are excellent for math or science teachers. However, teachers in other curricular areas usually are not accepted. The great need for introductory, nontechnical courses in educational computer technology for all teachers is still largely unmet. At Oregon State University, however, NSF is sponsoring the development of such a course, packaged as a multi-media introduction to computer concepts, the educational uses of computers and the cultural implications of automation. When completed, the course will be available for dissemination to schools or colleges.

In C. I. NETWORK the first course offered for all teachers was in computer appreciation, an introduction to computers and data processing in education. Many teachers did not need to take further course work. However, teachers who planned to teach about computers, teach programming or use the computer as a problem solver or curriculum extender often elected to go on to the next offering, Introduction to Computer Programming. This can be a one- or two-quarter course. It seemed to be less harrowing if the content were taught in two separate courses: Machine and Assembly Language Programming and Macro-Language Programming (FORTRAN and BASIC). Teachers could skip the Machine and Assembly Language course if they planned to use only a Macro Language such as FORTRAN, BASIC, ALGOL or COBOL for problem solving. The actual programming courses require access to a computer or terminal.

In addition to the three sequential classes mentioned above, C. I. NETWORK periodically offered one- or two-day seminars for teachers in a particular subject area. For instance, a Saturday seminar introduced mathematics teachers to BASIC and time sharing, and gave them a chance to experiment with applications from the classes they were teaching. A Teletype terminal gave access to a distant computer. Another half-day session for guidance counselors provided information about data processing jobs, careers and college and university offerings. A two-day "handson workshop" for all teachers was organized into large group lectures and small group laboratories, providing all participants experience in writing programs and using a computer or terminal. These activities can serve as "teasers" for the main event, the formal classes.

The shortage of appropriately trained teachers--indeed, the utter lack of such people--can be blamed in part on the colleges and universities. Not only are there no appropriate courses for undergraduates in most schools of education, but appropriate graduate level courses in any college are difficult to find. In most universities, both prospective and current teachers are shunted into the courses offered for professional programmer training or are given a half hearted "quickie" course in FORTRAN. Often these courses only discourage or divert teachers from studying computers again. The special needs of teachers are usually ignored or even rejected as not being practical. The fact remains, however, teachers need to understand the why and how of computers if they are to be confident in teaching or using computers. More time spent on fundamentals and less on the complexities of a particular scientific programming language would provide more adequate preparation. In addition, teachers and administrators alike need concrete information to help them make the decisions they are increasingly called upon to make about computer instruction.

Certain topics should be explored in an introductory course to prepare teachers for decision making and planning a computer instruction program. These include:

Types of computer hardware available: suitability, cost, selection criteria

Some languages currently available: the best application of each one

Time sharing: costs, advantages and disadvantages

Published and audiovisual resources available

Teaching methods

The various educational uses of computers

Whenever possible, teachers should be given time to see, use and evaluate hardware, software and resource materials.

An introductory course in computer appreciation, then, should include the following elements:

Nature of a computer

Organization

Operation

System components

Use of machine level and Macro Languages in computer programming (introduction only, using the simplest and most straightforward languages available)

Educational uses of computers

Administrative record keeping and data processing

The computer as the object of instruction (computer appreciation and business or vocational training)

Mathematical and scientific problem solving

Computer extended instruction: examples in every curricular area

Computer prescribed curriculum

Computer assisted instruction

Understanding computer instruction methods, equipment, languages and materials (including programmable calculators, portable computers, time sharing terminals and various input/output devices).

Social and cultural implications of the automation age

Teachers ordinarily will not be able to write programs or use the computer effectively until they have further course work in programming.

Experience has shown that teachers who first take a programming course do not develop a comprehensive understanding of computers. This is particularly true if the programming course is FORTRAN. A knowledge of FORTRAN seldom includes an understanding of computers, their limitations, capabilities, uses, and--in particular--their most appropriate applications in education. In fact, the teacher who is knowledgable about FORTRAN has misplaced confidence and bravado. Since he usually is the only teacher in the district with any computer training, he often makes decisions affecting curriculum and budget.

LANGUAGES FOR COMPUTER INSTRUCTION

The selection of programming languages for instruction probably is more important than the selection of the hardware. The confusing potpourri of languages available often prompts a hasty decision to adopt the most familiar language or to select a computer first and then use the languages in the accompanying software package. However, use of the wrong language actually can hinder, rather than help, in the achievement of objectives.

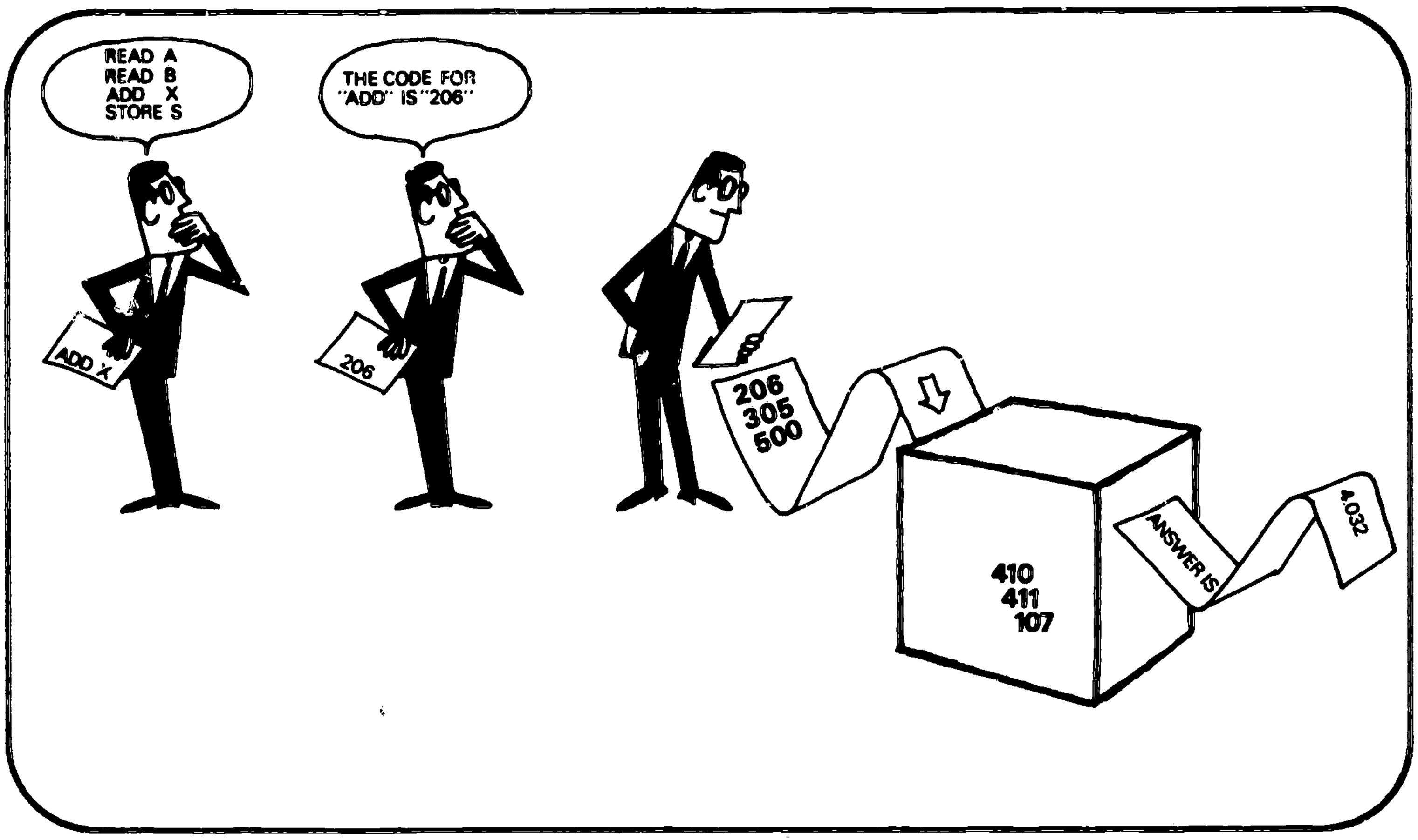
MACHINE LANGUAGE

In the early days of the computer age, all programming was done in what is referred to as "machine language." Machine language is the lowest level of programming language available, and the closest to the computer itself. Numerical codes are used for all instructions given to the computer, rather than words or other alphabetic symbols.

Every computer is wired to interpret and execute a set of simple operations, such as "add," "subtract," "store" and "read." Each of these operations has a unique numerical code. For instance, some of the operations for a hypothetical computer might be:

Operation	Code
Add	1
Subtract	2
Store	3
Read	4
Stop	5

For this computer to solve a problem, the programmer must prepare a detailed list of instructions using the set of operations supplied with the computer. He then would have to "code" each instruction, writing the appropriate numerical code for each operation and storing the entire list of numbers (which actually are codes for computer operations) in the computer memory. Once the list of instructions, called a "program," was stored in the computer memory, the computer could be told to execute the entire program. The computer would examine each instruction in turn and perform the indicated operation.



Computers, basically, understand only numbers. Therefore, the entire program must be put into memory in numerically coded form. This places the burden of translation on the human being, the programmer. He must translate all of his English language instructions into numbers which the computer is wired to interpret. There is a high possibility for error in coding, copying and entering the program, and the programming process can be very tedious if the problem to be solved is complex.

ASSEMBLY LANGUAGE

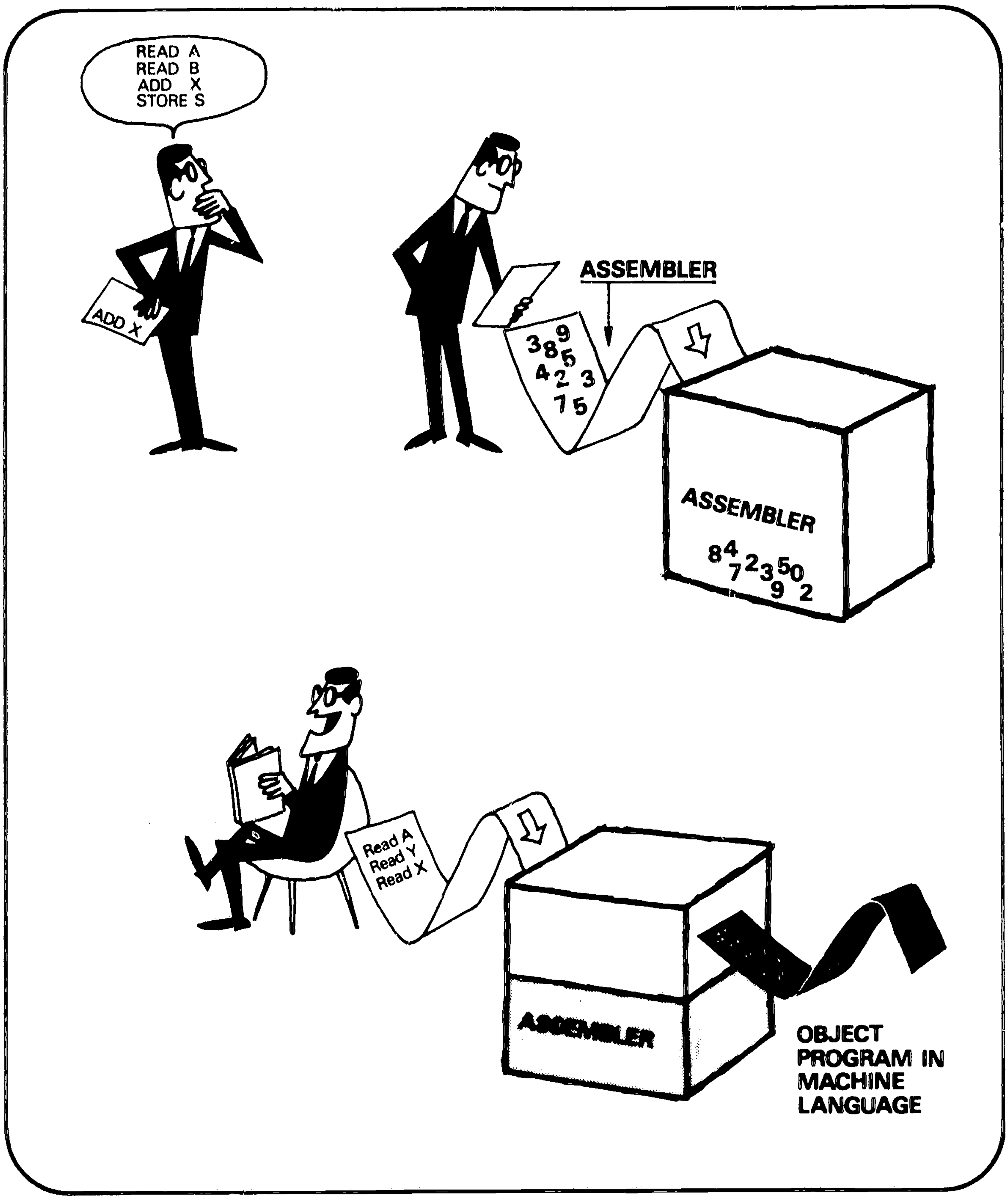
A programmer with writer's cramp probably realized, back in the early days of computing, the ability of the computer to manipulate symbols could be used to relieve some of the programmer's burden. It seemed logical that if a computer could be programmed, it could be programmed to do the tiresome "coding" of instructions like "add" and "store." So the next higher level of programming language developed was "assembly language." A programmer writing a program in an assembly language simply would plan and construct his list of instructions, using abbreviations rather than numeric codes for each operation. When the list of instructions was entered into the computer, a previously stored "assembly" program would examine each instruction and "look up" the appropriate code in a table stored in memory. This produced the ultimate machine language program, which then would be executed. The machine language program shown above, if written in assembly language, would appear as:

```
READ  X
READ  Y
ADD   A
SUB   B
STORE C
STOP
```

Since all programs executed by any computer must first be stored in the machine language, or numerically coded form, the above program would be translated by the assembler into:

```
410
411
107
206
305
500
```

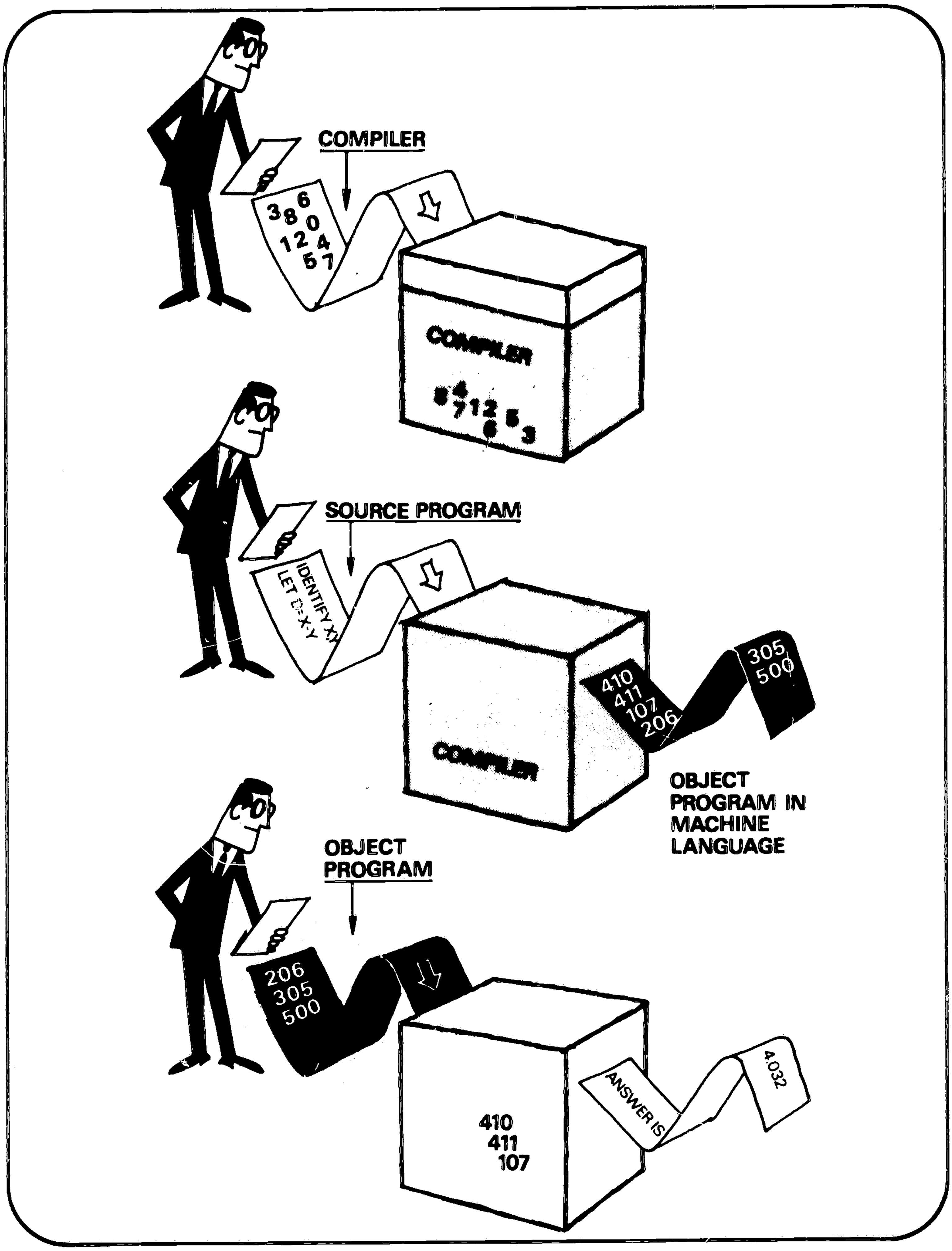
The process of "assembling" a program written in the symbolic form is diagrammed on the next page.


READ A
READ B
ADD X
STORE S
ADD X
ASSEMBLER
ASSEMBLER
Read A
Read Y
Read X
ASSEMBLER
OBJECT
PROGRAM IN
MACHINE
LANGUAGE

COMPILERS

Programmers soon realized, however, that only minor tasks had been taken over by the assembler program. They still had to write every single instruction themselves. In fact, just to perform the calculation "y = ax + b" could require from 9 to 30 instructions (depending on whether the computer came equipped with an automatic "multiply" operation; if not, a special set of steps to multiply two numbers would have to be part of the program to perform the above calculation). Whether the program was written in machine language and coded by the programmer, or written in assembly language and coded by the computer, the time spent on problem analysis was the same, and the same number of instructions had to be written. At this point, the super-translator was conceived: the "compiler." If a computer could be programmed to code instructions on a one-to-one basis, why could it not be programmed to accept an entire statement for "y = ax + b," analyze it according to some predetermined rules and produce the ultimate machine language program?

The development of the "macro language," or compiler, was the beginning of a new era in the still new age of computers. A programmer no longer had to painstakingly write out each single operation to be performed by the computer. Now, an entire formula could be written as one statement, to be computer translated into a corresponding list of operations.


COMPILER
386
0
124
57
COMPILER
SOURCE PROGRAM
COMPILER
305
500
410
411
107
206
OBJECT
PROGRAM IN
MACHINE
LANGUAGE
OBJECT
PROGRAM
206
305
500
410
411
107
ANSWER IS
4.032

The compiler language FORTRAN (FORmula TRANslation) was one of the first and still most widely used macro languages. Other languages soon were developed for special purposes. While FORTRAN was an excellent language for the solution of mathematical problems, it was not well suited to business applications involving few arithmetic calculations but a high volume of special format input and output. Thus COBOL (COmmon Business Oriented Language) came into being. Other popular compiler languages include ALGOL (ALGOrithmic Language), and recently, BASIC and PL/1. A simple program written in several languages would be:

ECP-18 (a machine language program)		ECP-18 (an assembly language program)
Memory cell	Instruction or data	
10	07 0023	/ORG 0010
11	15 0026	BRG C
12	14 0212	STO X
13	14 0215	A OIR 212
14	07 0026	OIR 215
15	02 0100	BRG X
16	16 0024	OAR X
17	15 0026	ADD C+1
20	17 0025	STO X
21	11 0012	SUB C+2
22	00 0000	JON A
23	00 0000	HLT
24	00 0001	C/BLO 000000, 000001, 000012
25	00 0012	X
26	X	/END

A program to print a list of the first 10 digits (0-9):

FORTRAN (a macro or compiler language)

```
   X = 0
 6 PRINT (4,15) X
15 FORMAT (F1.0)
   X = X + 1
   IF (X - 10) 6, 25, 25
25 STOP
```

BASIC (a macro or compiler language, but "conversational" in nature)

```
10 LET X = 0
20 PRINT X
30 LET X = X + 1
40 IF X<10 THEN 20
50 END
```

Note the computer for which the machine language program was written uses only "base 8" numerals; the digits 8 and 9 do not exist in base 8. Nearly all compilers (and some assemblers) will handle the conversion to base 10 automatically.

The most readily understood program of the four shown, and also the simplest to write, is the BASIC program. BASIC is a language representative of the recent trend toward "conversational" languages, which are similar to familiar English language and relatively free of special codes, format restrictions and control characters. Conversational languages such as BASIC are used most often at a remote terminal, usually a Teletype keyboard. FORTRAN is available now on some computers in conversational mode (notably QUIKTRAN), which means it can be used from a remote terminal and does not have the form and format requirements of a standard FORTRAN.

An even simpler version of the illustrated program can be written in either FORTRAN or BASIC, as shown below:

FORTRAN

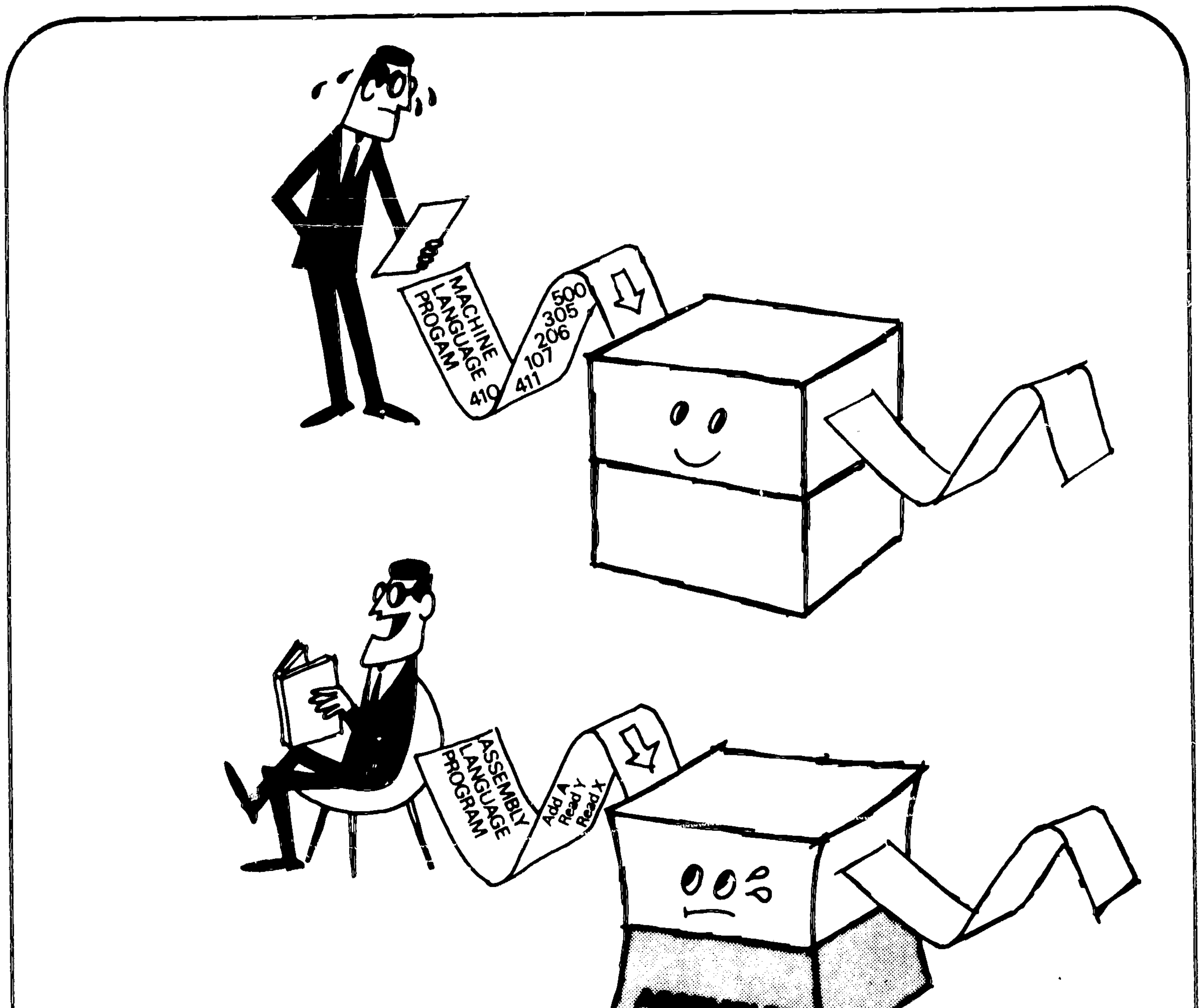
```
   DO 6 N = 0,9
 6 PRINT (4,15) N
15 FORMAT (F1.0)
   STOP
```

BASIC

```
10 FOR X = 0 to 9
20 PRINT X
30 NEXT X
40 END
```

By comparing the 4-line BASIC program shown above to the 16-line machine language program which accomplishes the same result, it becomes easy to understand why much of today's routine programming is done in some high level language such as FORTRAN or BASIC. The use of a compiler does simplify problem solving. However, most of the realities of the computer are masked by the sophistication of the language. In fact, the more sophisticated the language, the farther removed is the user from the computer, and the less likely he is to understand anything about its true capabilities and limitations.

Students debugging a machine language program on the ECP-18 computer



The programmer's task is simplified by using sophisticated languages which the computer must translate

A few simple programs, even shorter than the one shown, can be extremely enlightening to the student if he has an opportunity to write them in machine language and, most importantly, to try them on a real computer. The numbers and symbols written on a program sheet often are more confusing than instructive, unless the student loads them into a computer and tests the program. Students actually can watch the mindless, sequential, blind following-of-instructions, watch the transfer of information within the computer, and then truly can understand what a computer can and cannot do and how it "solves problems." They see the relationship between hardware and software (the machinery and the programs). Later, when a higher level language is introduced, students fully appreciate what has been gained by the use of a compiler and they can visualize realistically the compilation process. On the other hand, if students are taught a language such as FORTRAN or BASIC, but not allowed to get their hands on a computer, unfortunate misconceptions and myths are only reinforced. The Teletype terminal is often thought of as "the computer," and students have no real picture of what is at the other end of the telephone line. If programs are punched in cards and delivered to a computer center to be processed, the computer itself is an even more remote concept. Because of the power of the programming language and the remoteness of the computer, the idea that "computers are mysterious, awesome, omnipotent, thinking creatures" only becomes more firmly established in the mind of the student.

ORDER OF INSTRUCTION

Arguments have continued for years about the best time to introduce a study of machine language. It can occur after students have learned a high level language or before other instruction takes place. In the Computer Instruction NETWORK machine language was introduced first, followed by a brief exposure to assembly language and finally BASIC or FORTRAN. Control groups practiced the opposite order, but teachers of these classes soon expressed their preference for the machine language-to-compiler approach. The teachers observed that students who first learned a language such as BASIC or FORTRAN and were accustomed to the freedom of using such a language did not want to progress "backwards" into a machine level programming language. Machine language requires more effort to produce the same results and appears to be much more limited than the compiler language. Students saw no justifiable reason for studying it when higher languages were available. The real purpose behind such study--to develop an understanding of the true nature of the computer--was lost in the protest.

Some students became entranced with the intimate machine language approach to the computer and never came out of it. Every computer teacher recognizes the "computer whonk" who reports in before the janitors every morning and hides in a closet so he can spend the night with the computer. This student often prefers machine language over any other, and becomes over involved with elegant (but sometimes trivial) programs and button pushing. This can happen regardless of when machine language is introduced and is probably an unavoidable consequence. Students experience a feeling of excitement and unusual power when they can communicate successfully with a machine and motivation is consistently high. It is natural for some students to get "hooked" by the computer and want to prolong the relationship.

Perhaps a solution to the dilemma of the order to be used is one which proved highly successful in some NETWORK schools. Students were given an "appetizer"--a brief demonstration of the power of the computer and introduction to programming in a language such as BASIC. Once their interest was stimulated, a closer examination of the computer was in order. At this point the teacher introduced fundamental computer concepts, including machine language programming. From there students progressed to the use of a more sophisticated language for problem solving and exploration.

SELECTING A LANGUAGE

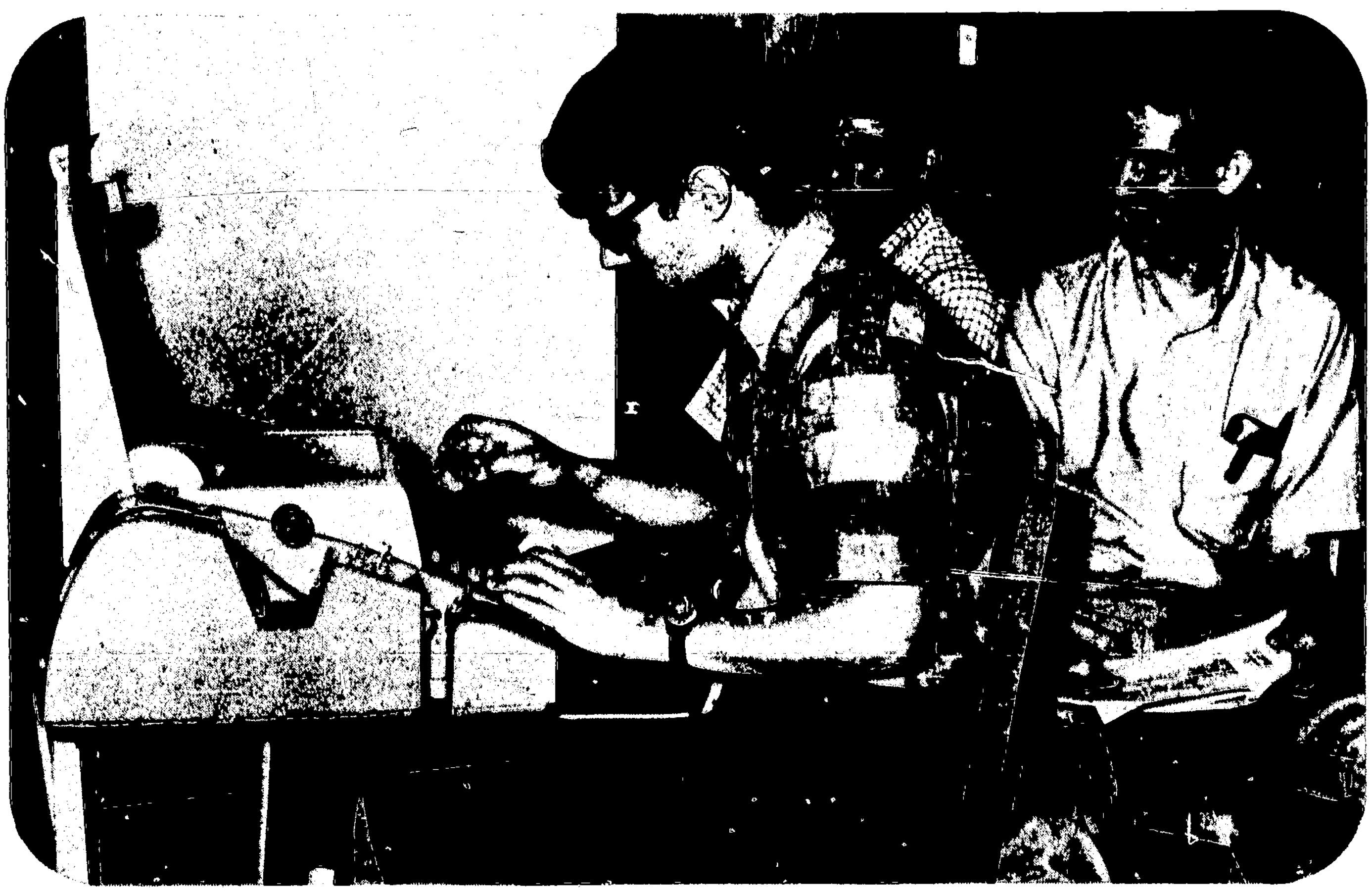
The original objectives of the planners must provide the criteria for selection of a language. They must determine if the computer will be used for a mathematics laboratory, scientific problem solving, business education, prevocational training in programming and machine operation, or computer appreciation. The last two applications should include a study of machine language. Most computers available today boast complicated instruction repertoires and are not designed for instruction or ease of programming and operation. If such a computer is the only one available and fundamentals are to be taught, a subset of the machine language should be identified for instruction. A teacher or other comparable person should be given time to specify this subset and to prepare appropriate teaching materials.

Barring this alternative, a simulated computer may be used. A teacher may contrive a "blackboard computer" with a limited number of memory cells and instructions, and may take the class through some simple programs in this way. A cardboard simulated computer called CARDIAC also is available. Motivation will not be as high without handson experience and understanding will necessarily be more limited. However, an imaginary computer might be preferable to immersing the entire class in the complexities of a machine language apparently designed to be impenetrable.

If the instructional program is heavily oriented toward vocational training, a student should have experience with programming languages from machine to assembly and compiler level. An understanding of machine language is a valuable prerequisite to the study of all higher level languages and greatly reduces the time required to learn them.

MATHEMATICS AND PROBLEM SOLVING

Many mathematics teachers believe the computer should be used only as a tool in their classes and never as the object of instruction. Used



A mathematics laboratory

in this way, as a mathematics laboratory, the computer extends instruction in mathematics. Students may investigate mathematical concepts with the aid of the computer, may use it as a tool for previously impossible calculations, or the teacher may use it to demonstrate selected concepts. There is another benefit to be derived from the use of a computer as a mathematics laboratory, which is perhaps more to the point. Dr. Virginia T. Gilbert has stated three levels of understanding in mathematics. The first level is achieved when one hears a concept explained. The second comes in attempting to explain the concept to someone else. The third and highest level of understanding is gained in explaining the concept to a computer. This has been borne out in NETWORK classes. A correct result will be achieved only after a complete and logically organized program is developed. To instruct the computer on the logical method of solving a problem, the student must clearly understand the method himself. It is impossible for a student to follow the example given in his mathematics book in a "cookbook" manner in developing a general algorithm for a computer program. He must understand the concept, be prepared to consider all possible exceptions and know the parameters. He also must be aware of several different methods for solution, since the computer might operate more efficiently on an algorithm not best suited for pencil and paper solution.

Used in this way, the computer can assist in developing understanding of mathematical concepts, as well as in developing logical techniques for problem solving. There are other, possibly more important, ways the computer can be used as a mathematics laboratory. At any rate, if mathematics and science teachers wish to make use of the computer as a tool for exploring and solving mathematical and scientific problems, they probably are not inclined to devote class time to any study of the computer itself. Whether or not they should is an often debated point. If social studies teachers or a team of teachers cannot or will not offer a computer appreciation course, the task often falls to the mathematics teacher who has shown the most interest and inclination toward "teaching computers" and sometimes he rebels with justification.

The task of choosing a language and a computer for problem solving becomes less formidable if one considers the objectives. The problem to be solved and the method of analyzing that problem are the important instructional objectives. As little time as possible should be spent in teaching or learning the minutiae of a computer language. For students to solve scientific problems of any meaningful complexity, they would have to learn enough machine language and programming techniques to qualify them as quite competent programmers. In addition, they would have to be well versed in the idiosyncrasies and unique characteristics of the particular machine they were using. The time spent in this activity necessarily would detract from the objective: to use the computer as a problem solving tool. In fact, students would often be more absorbed in manipulating the computer language than in solving the problem, and a proportionately small amount of time would be spent on the problem itself, as opposed to the writing of the program.

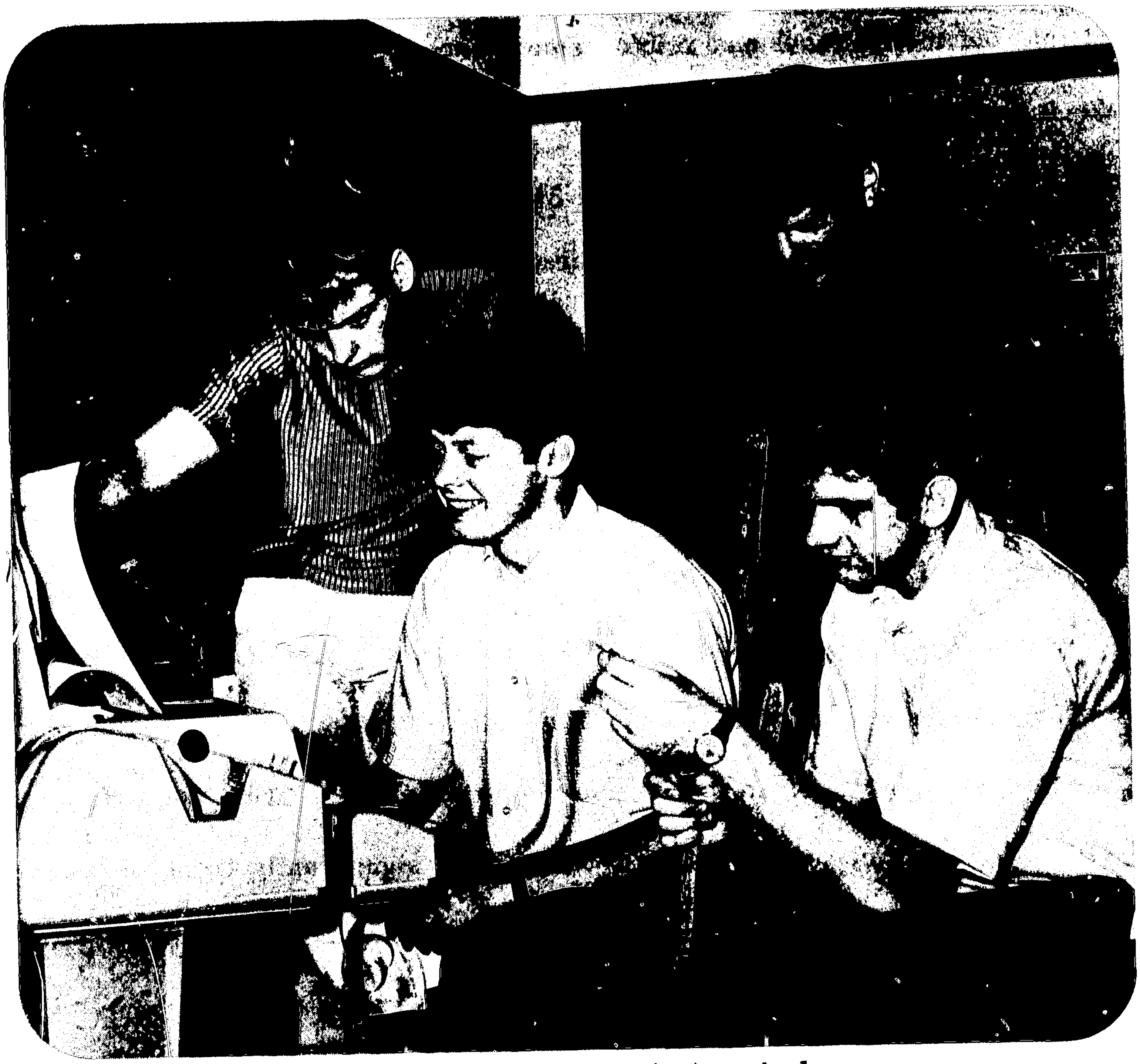
By the same token, however, available macro languages, or compilers, must be examined if they are to be used in problem solving. Most commonly used macro languages impose equally undesirable restrictions, although different from machine language. FORTRAN and ALGOL, for instance, require rote memorization of many otherwise meaningless symbols. In one inservice class an entire three-hour-session was devoted to demonstrating to teachers the many ways in which a slash (/) could be used and interpreted in a FORTRAN program. Although these languages are admittedly powerful in handling mathematical and scientific problems, much time must be devoted to details unrelated to the problem to be solved. One can get a feeling for this merely by examining the FORTRAN program on page 35 and comparing it to the one written in BASIC.

On the other hand, the newer conversational languages eliminate the need to learn much about a language other than a set of very simple instructions such as "PRINT" or "INPUT." There is no need to differentiate between real numbers and integers, no "special" but otherwise meaningless symbols required by the computer, and no complicated formatting for input and output. The need for rigorous format restrictions can be questioned, since a mathematical or scientific problem seldom requires an elegant input or output format. Often output consists of a single number of a simple list of numbers.

Conversational languages, then, offer an ideal tool for problem solving. Some experts advocate teaching a programming language only to the extent it is needed to solve the problem at hand. The simpler the language to learn and use, and the closer it is to natural speech and thought, the better suited it is for problem solving.

It should be noted a simple language is not necessarily less "powerful" than a more complex language. BASIC is in many ways as powerful as, and in some ways more powerful than, FORTRAN. Certainly it is sufficient for most problems being explored by a high school mathematician. If the magnitude or format of a problem being solved exceeds the ability of the compiler to handle it, most time sharing systems have available a selection of languages of varying complexity and the user may switch to any other languages instantly. If this requires the student to learn the details of a new language, it is not as wearisome a task at this point. Fred Gruenberger has quoted a "5% Law" which states that "any new computer language may be learned in 5 percent of the time it took to learn the first." This was borne out in NETWORK classes.

The selection of a simple but powerful language for problem solving, then, usually means that a time sharing service must be acquired. That is, the school must lease from the telephone company a Teletype or other keyboard device connected to a telephone line, and sign a contract with a commercial or university time sharing system. This contract allows the school, for a price, to dial the large, remote computer and enter programs from the terminal. They are run immediately and the results printed out in the classroom.



The computer prints results at a remote terminal

Most computers owned or leased by school districts do not have conversational languages available. In fact, the very definition of the word "conversational" implies some sort of interaction between user and computer. Certainly the motivational and reinforcement aspects of online communication make it far more desirable for problem solving, particularly where the student is exploring some problem with the goal of discovering relationships or mathematical concepts. It is discouraging to a student to submit a program which he is convinced is adequate, only to discover the next day or a few days later he has received from the computer a list of "error messages" or an obviously wrong answer.

BUSINESS EDUCATION

While the ideal mode for scientific problem solving seems to be the use of conversational languages at remote terminals, this may not be the ideal mode for instruction in a business education class. The objectives in an office practices class, for instance, may vary significantly from school to school. In one class, the teacher may wish to use the computer much as a mathematics teacher might, as a tool to aid in the solution of problems encountered in the class. However, he might be more interested in solving the problem in a way which is typical of methods in businesses. Students may be required to learn FORTRAN, since this is a commonly used language in business computer installations, and may keypunch their own programs before submitting them to the computer. Or, if the COBOL language were available, it would be equally representative of a typical business installation. COBOL, in fact, is simpler to learn than FORTRAN, although more hardware is required to run COBOL programs than FORTRAN. COBOL makes use of magnetic tapes (often used for storing and retrieving "files") and other auxiliary storage and input-output devices which may be optional in a system which can run FORTRAN programs. Therefore, if a business education teacher wishes to teach the commonly used business language, COBOL, he must first make sure a system equipped to handle COBOL programs is available nearby. It is unlikely COBOL capability can be found in a time sharing system--at least not for remote terminal users. It may be used in a time sharing system for those "background" jobs which are run only at the computer center when the computer is not busy handling remote users. Since a COBOL program usually will require the mounting of a magnetic tape or disk, or several of them, a computer operator should be available to supervise the running of student programs.

INSTRUCTIONAL RESOURCES

When selecting a language for instruction, an important criterion is the availability of appropriate instructional resources. The availability may depend on the extent of its current usage for instructional purposes. New "conversational" languages are being announced regularly, and some commercial vendors of time sharing services will insist their new conversational language is so superior it would be folly to choose another. But the question must be asked: who else is using it? This is one time when being a "sheep" rather than a leader has definite advantages. A language which has been used by many other schools has been thoroughly tested in an educational environment. Other schools have complained about undesirable characteristics of the language, and perhaps the language has improved accordingly. But most importantly, published texts will be available. Manufacturer's reference manuals are nearly impossible to use for instruction. Students who are accomplished programmers often can use these manuals for reference, but they are undecipherable for beginners and only discourage them. It is essential, then, to find other sources for the classroom texts and instructional materials. An unknown language may be a very good one, but associated teaching materials probably were developed by technicians who did not understand instruction or communication. The result is yet another incomprehensible reference manual. Languages such as FORTRAN, COBOL, ALGOL and BASIC have been in common use long enough that a number of commercial textbooks are available.

OTHER LANGUAGES

Besides the ubiquitous FORTRAN and BASIC, several other languages are frequently mentioned as good choices for instructional use. COBOL is relatively uncomplicated to learn and use, but requires a diversity of hardware usually found only in a business installation.

ALGOL (ALGOrithmic Language) is a more powerful and complex language than FORTRAN, BASIC or COBOL. Besides being a programming language and translator (or compiler), it also is designed as an internationally accepted procedure for designing mathematical, engineering and scientific problems. ALGOL is available on some systems in conversational mode, along with BASIC and FORTRAN. If a teacher or student wishes to use the unique characteristics of ALGOL, he may learn the language with relative ease after having used BASIC. It is much easier to learn more complex languages if one has first become familiar with the basic concepts of computers and programming using a simpler language.

QUIKTRAN is a version of FORTRAN, modified for conversational use with remote terminals. It is similar to BASIC in this mode, but is encumbered by more special meaning symbols and terminal commands. While QUIKTRAN may be used for online program construction and debugging, it also may be used in a "batch processing mode." In this mode, programs are processed offline at a time when the system is not busy with conversational operations. There is no one-for-one communication between the user and the system. The statements are collected from the terminal and saved on an auxiliary disk storage device for later processing.

Programming Language/One (PL/1) was introduced with IBM's System/360 computers in 1964. PL/1 narrowed the gap between the computer needs of commercial and scientific users. The language can be used for many applications in both areas, including real time processing, systems programming, Teleprocessing and command control. PL/1 is capable of handling the most complex computing problems. Thus it is equipped with an almost endless list of possible commands and special characters. The modular design makes it possible to teach only a part of the language to beginners, permitting them to write programs. On the other hand, the advanced programmer can take almost unlimited advantage of the computer's versatility as the need arises.

Because of its complexity, however, special control characters and commands must be used to program from a terminal. The language is highly flexible and powerful for the professional programmer, but was not designed for instruction. Postsecondary vocational schools and data processing schools, nonetheless, tend to choose PL/1 as one of the languages to be taught because of the pervasive influence of IBM and their 360 computers.

In the December 1968 issue of Datamation magazine, however, the editor quotes an independent survey which found only 1 percent of all U.S. installations were programming in PL/1. This indicates somewhat less than "universal acceptance" to date of the language which was deliberately designed as a "universal" language. In a comparative language study recently done by Logicon for the Air Force, PL/1 was the unanimous choice of professional programmers for their own use. At the same time, however, there was almost complete agreement that COBOL was more appropriate for the nonprofessional in the business area and FORTRAN for "other" nonprofessionals. The programs compared in the study included "interactive programming" (development of an online system), simulation and gaming, and scientific problems. Use of a programming language in an educational environment was not considered in the study, so the results do not necessarily indicate that FORTRAN is the ideal language for nonprofessional problem solvers in the classroom.

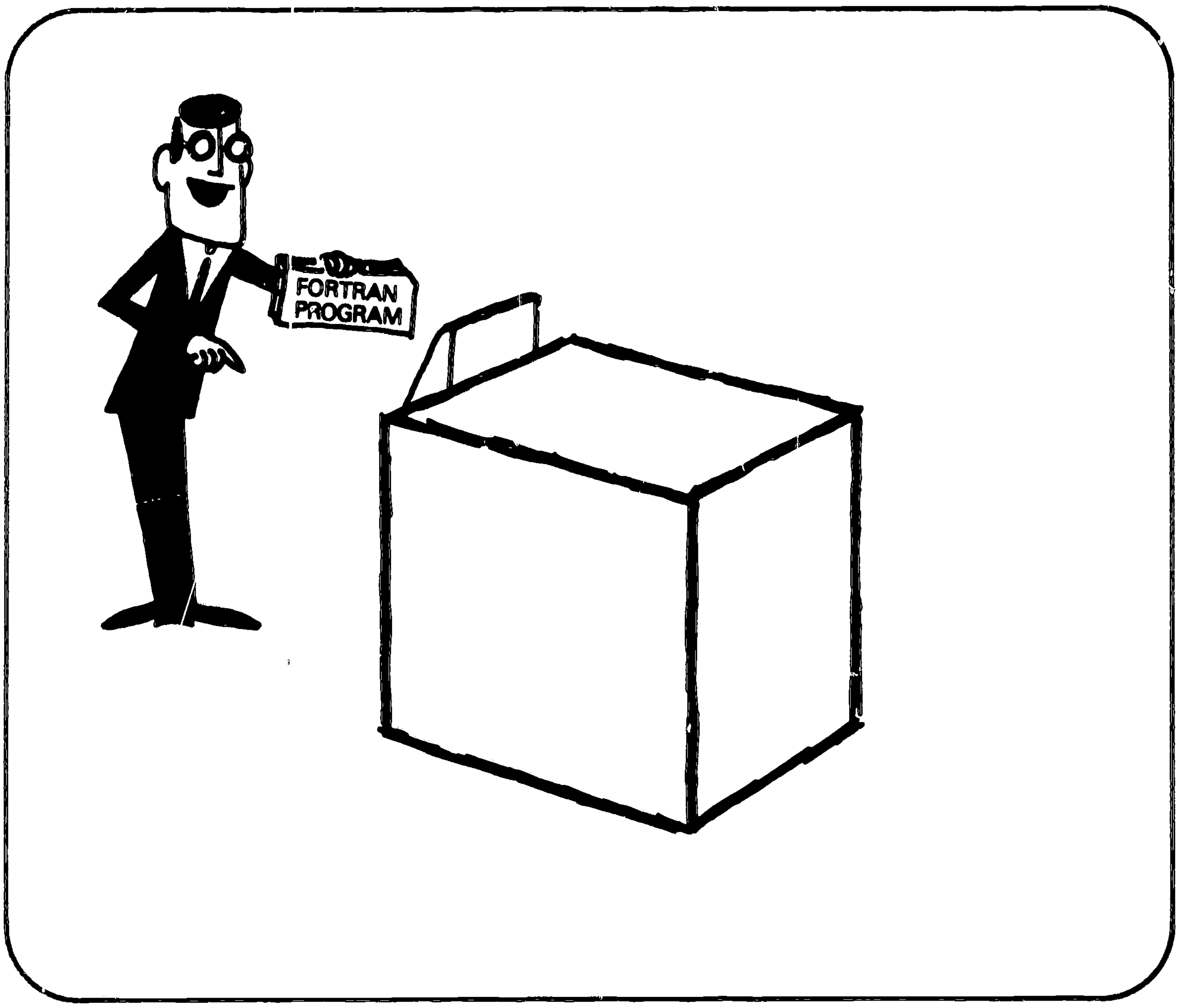
C. I. NETWORK LANGUAGE SELECTION

Computer Instruction NETWORK curricula included, in most cases, a brief study of machine language programming with handson operation of the computer by students. After each student had attained a minimum understanding of computers, a time sharing terminal was made available and students learned to program in BASIC.

One of the small computers used for classroom instruction in machine language was the ECP-18, an instructional computer with a limited memory and no compiler language. The other was the PDP-8/S, which does have FORTRAN available. However, the steps required to compile (translate) a program and the time involved made FORTRAN impractical for classroom use on the PDP-8/S. In 1968, however, Digital Equipment Corporation introduced a compiler called FOCAL, which allowed the user to enter a program written in FOCAL and to have the program instantly compiled and executed with no intermediate steps. That is, compilation took place internally and the resulting machine language program was automatically stored and made ready to run. In operation, then, FOCAL on the PDP-8/S was nearly identical to BASIC on a time share terminal. The existence of FOCAL made it easier for NETWORK staff to write "CINET BASIC," or BASIC written for the PDP-8/S, using FOCAL subroutines. It was then possible for all NETWORK participants to continue to use the language (BASIC) they had learned and collected materials for, even without a remote terminal.

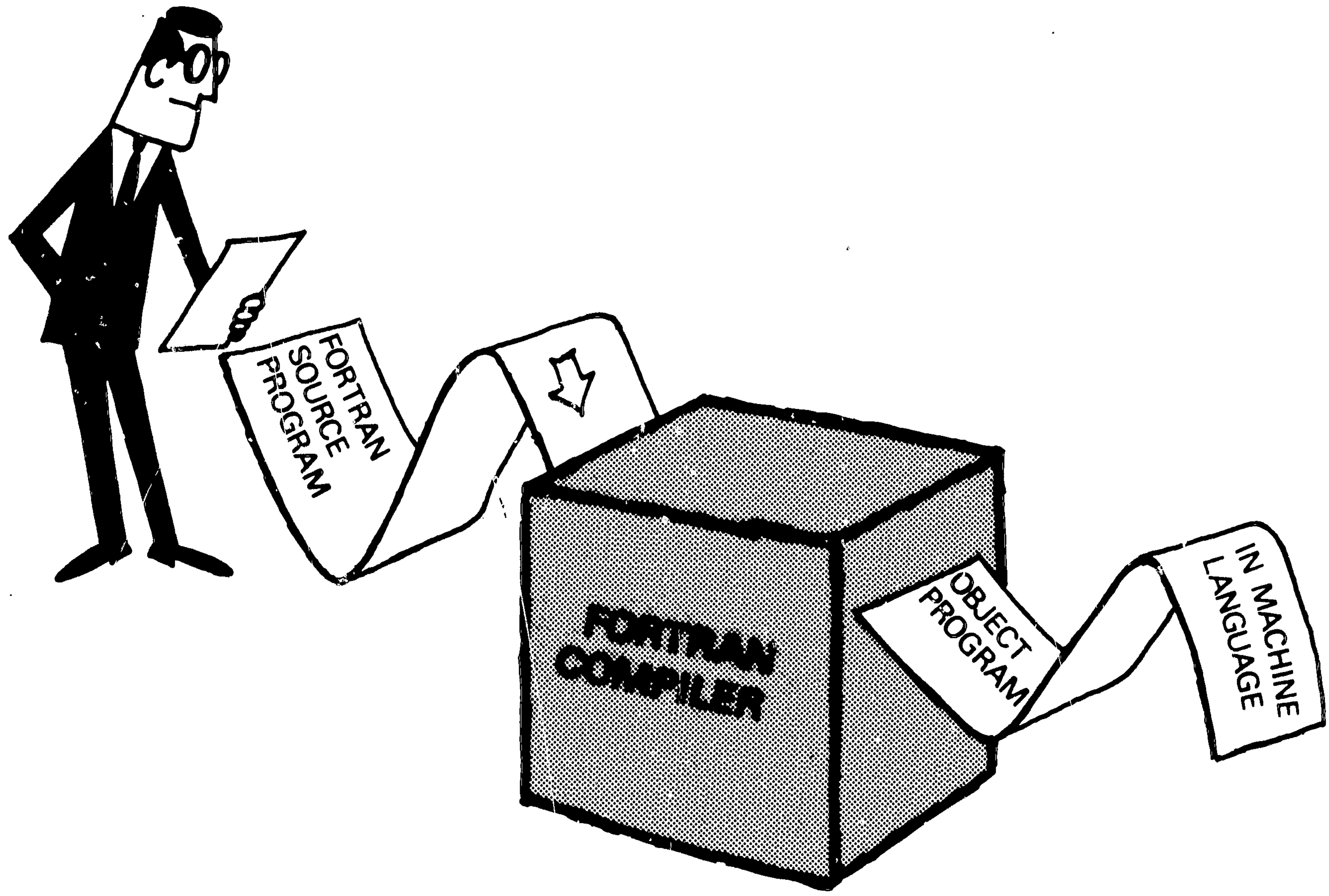
While the NETWORK planned to develop BASIC for use with the IBM 1130 in the Computemobile (the NETWORK's truck mounted unit), it primarily was used to run FORTRAN programs. The 1130 had online disk storage for the FORTRAN compiler, so the compilation process appeared to be as "instantaneous" as with CINET BASIC on the PDP-8/S. This "instant compilation" was vital if student programs were to be run during the class period. The only alternative was for students to submit their programs to a central computer for processing at a convenient time, with results returned to the class in a day or more.

A word about compilation will clarify this. Typically, when a computer user writes a program in FORTRAN, he must prepare the "input media" for entering the program into the computer. This usually means he must keypunch a deck of cards with a FORTRAN statement on each card or punch a paper tape. Then he uses the deck of cards or roll of tape to "load" the information into the computer. He may skip a step by using the console keyboard or a remote terminal to enter the program directly.

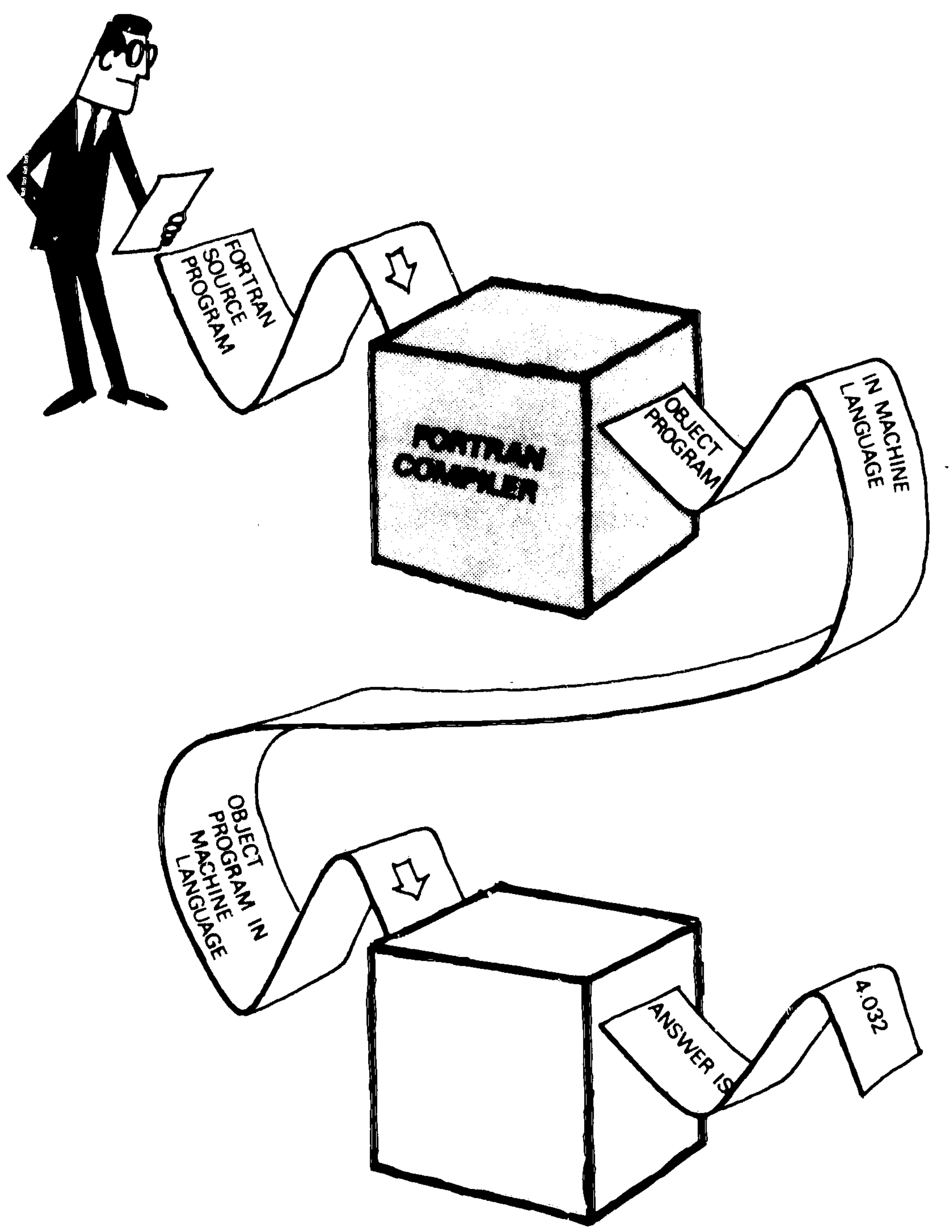


The computer user punches his FORTRAN program on cards to load into the computer

As the FORTRAN program, called the "source" program, is read into the computer, the compilation process begins. At this point the capacity of the computer's main storage and the availability of online auxiliary storage can make a crucial difference in compilation time. The compiler (or translator), which actually is a program, must be stored in memory before the incoming source program can be translated and the resultant machine language program produced.



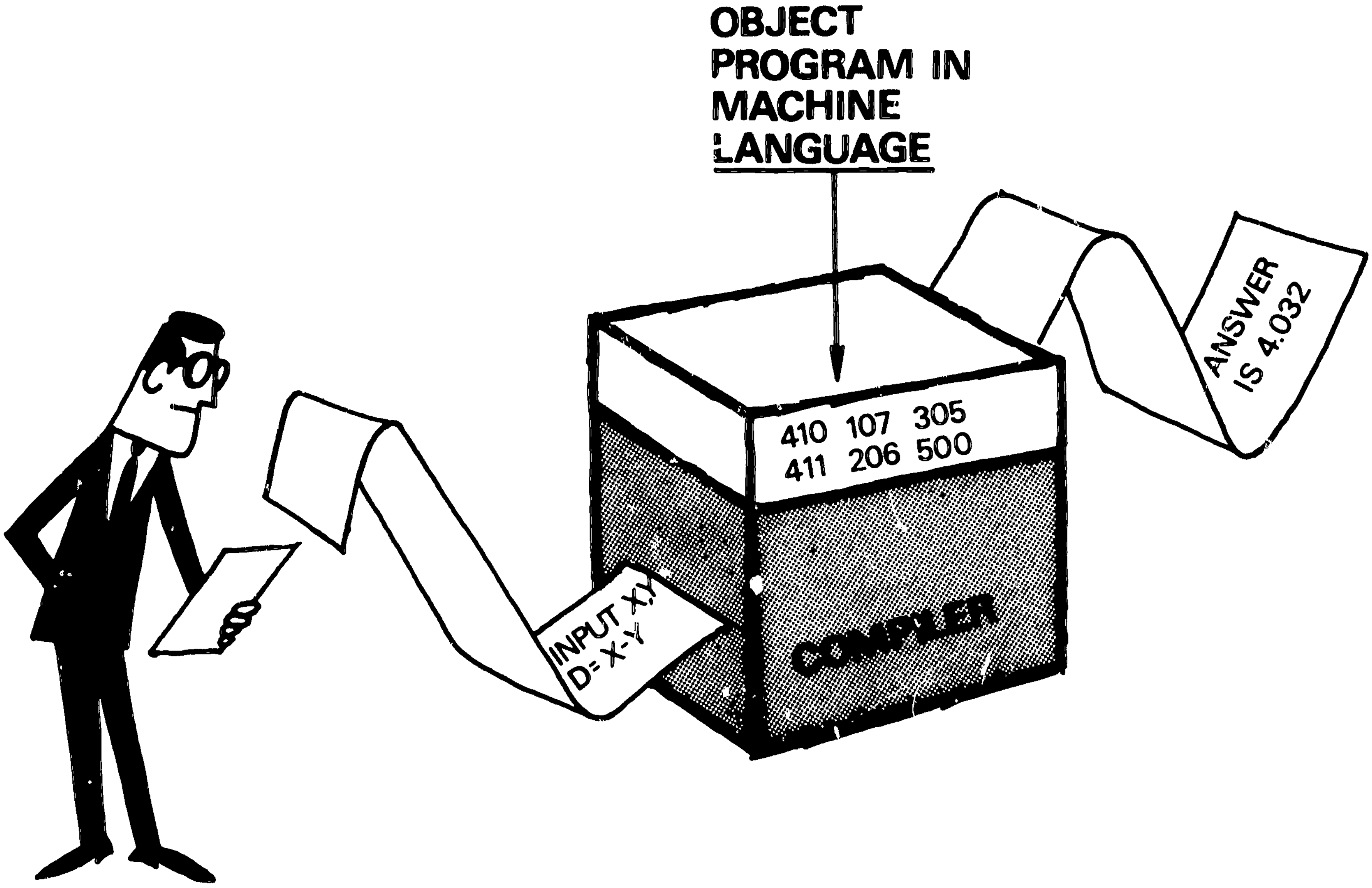
The PDP-8/S used in the NETWORK contains 4096 ("4K" in computerese) storage locations. Since most compilers or translators occupy more memory than this all by themselves, the PDP-8/S FORTRAN left no memory space for the compiled machine language program to be stored. Consequently, an intermediate tape called the "object tape" was punched in machine language by the computer, and the user was required to load the tape back into the computer, wiping out the compiler in the process.



For a computer system using punched cards but with limited memory capacity, the procedure would be essentially the same. The compiler program would be loaded into the memory; the source deck would be read; the compiler program would translate the source program and punch an "object deck" of cards in machine language; the object deck (machine language program) would be read into memory, in the process erasing the compiler program; and, finally, the program would be run. All of this, even for a short program, could take from 10 to 30 minutes, and the next student would have to start with the first step again.

For this reason, it is necessary to have available a "load and go" compiler. It must not occupy all of the memory and must translate and store the machine language program without punching an object program. Thus, the compiler stays in memory at all times, and each student need only enter his own source program and observe the results. Granted, the memory remaining for student object programs probably would be limited but if programs are not lengthy or complex, this is no hindrance. Better yet, auxiliary storage such as a disk would eliminate all compilation time problems.

The other major time problem is created by the input bottleneck. Students punching their own cards or tape, or typing in programs directly, spend valuable time doing yeoman's work. In addition, the time consumed in simply reading in the card decks or tapes, or in laborious typing, is time better spent by students getting their hands on the computer itself. This aspect is discussed in more detail in the section on equipment.


OBJECT
PROGRAM IN
MACHINE
LANGUAGE
410 107 305
411 206 500
COMPILER
INPUT X,Y
D=X-Y
ANSWER
IS 4.032

EQUIPMENT FOR COMPUTER INSTRUCTION

The wonderland of computer hardware becomes more esoteric every day as "third generation" computers replace older transistorized models. Schools faced with decisions about lease or purchase of computing equipment often are overly concerned about "obsolescence" and possibly less concerned with suitability. In selecting a computer for instructional purposes, again the instructional objectives must determine the criteria.

A business or administrative computer system will be selected by quite different means for quite different reasons. For example, in an administrative system output likely will be copious and highly formatted. Few one-time programs will be run, as most applications will be standardized and programs run on a regular schedule. Large data files will be maintained on auxiliary storage. Handson operation of the machine will be discouraged as inefficient; in fact, probably only a few duly authorized computer operators will be allowed to touch the machine.

An instructional computer, however, should be used for handson operation by students, for the running of many one-time programs and for input and output of unsophisticated format. Large storage areas and data files probably will not be needed. The standards, then, are not as high as for a business system. The computer is now a laboratory tool, and costs cannot be reckoned by the same methods used for a "production" system.

The differences in objectives and application of the business system and the instructional program cause problems any time a school district administrative computer is used for instruction and running student programs. It is not economical or efficient, or even smart, to allow students to get their hands on a system carefully designed for optimum operation. If a student accidentally erases a single magnetic tape file or master disk, it is unlikely instruction will take place ever again in the computer center. In addition, instruction must take place during prime time, during the day shift when most of the computer's tasks are being accomplished. It does not seem wise to dedicate some of that costly time to running student programs, even when students are not allowed to do it themselves.

The purchase or lease of a low cost computer or terminal to be used as a laboratory tool probably will prove to be most effective in solving this conflict, as well as being more economical in the long run. Not only is the district computer larger and more expensive than it needs to be to meet the needs of the instructional program, but it simply is not designed for ease of instruction or for laboratory use.

The equipment configurations discussed here, then, will not be the type used for administrative data processing in school district offices. They will be more appropriate for instructional objectives.

INSTRUCTIONAL OBJECTIVES

Do you want to teach about computers to develop a basic understanding and appreciation of the computer and automation? To truly understand the computer, handson operation of a simple, but real, computer is a valuable assist. Such a simple computer, however, can give students too limited an idea of the power of an information system. A simple computer for handson operation can be supplemented by access to a large and sophisticated computer system via a remote terminal.

For this same objective, a low cost computer equipped with a conversational compiler such as BASIC can serve both purposes. It can provide for handson operation in machine language and problem solving in a high level language. However, some compromise must be made in both areas. Such computers are not designed specifically for ease of instruction, and a machine language subset would have to be identified. Operation of the computer itself may not be as simple as it would be on an instructional device. In addition, a small-to-medium-size computer on site cannot have the capacity or power of a large time shared system accessible from a terminal. For the stated objective, though, this small, low cost machine should be adequate.

If the objective is to provide a problem solving tool in a mathematics laboratory, the most powerful computer with the least to learn about programming and operation may be the answer. This seems to indicate a remote terminal with an easy-to-learn conversational language.

For business education, the equipment desired can vary depending upon the particular course being taught. Many times a course in data processing turns out to be the rudiments of operating unit record equipment; that is, machines which manipulate punched cards in some way. Unit record equipment can be used without a computer to sort, collate, merge, tabulate and otherwise use punched cards as input. Each individual card is considered as a single "record," and the data contained on the cards is processed in the sense that it is used to produce new data (new cards representing updated records).

The machinery needed to teach operation of unit record equipment can cost as much each month in lease costs as several remote terminals or a very adequate computer. Although tab operators, or operators of unit record equipment, are still being hired, most such data processing operations are now handled more efficiently by computer.

Another practice of questionable merit is the high school course in keypunch operation. In one high school four 9-week courses were offered each year in keypunching, with 30 girls in each class. That high school was turning out 120 keypunch operators a year in a town where the total demand for this skill probably never exceeds 10 keypunch operators a year. The managers of large computer systems often prefer to train their own keypunch operators and require only that the girl can type when hired. The increasing use of remote terminals and direct keyboard-to-magnetic-tape devices is decreasing the use of keypunches.

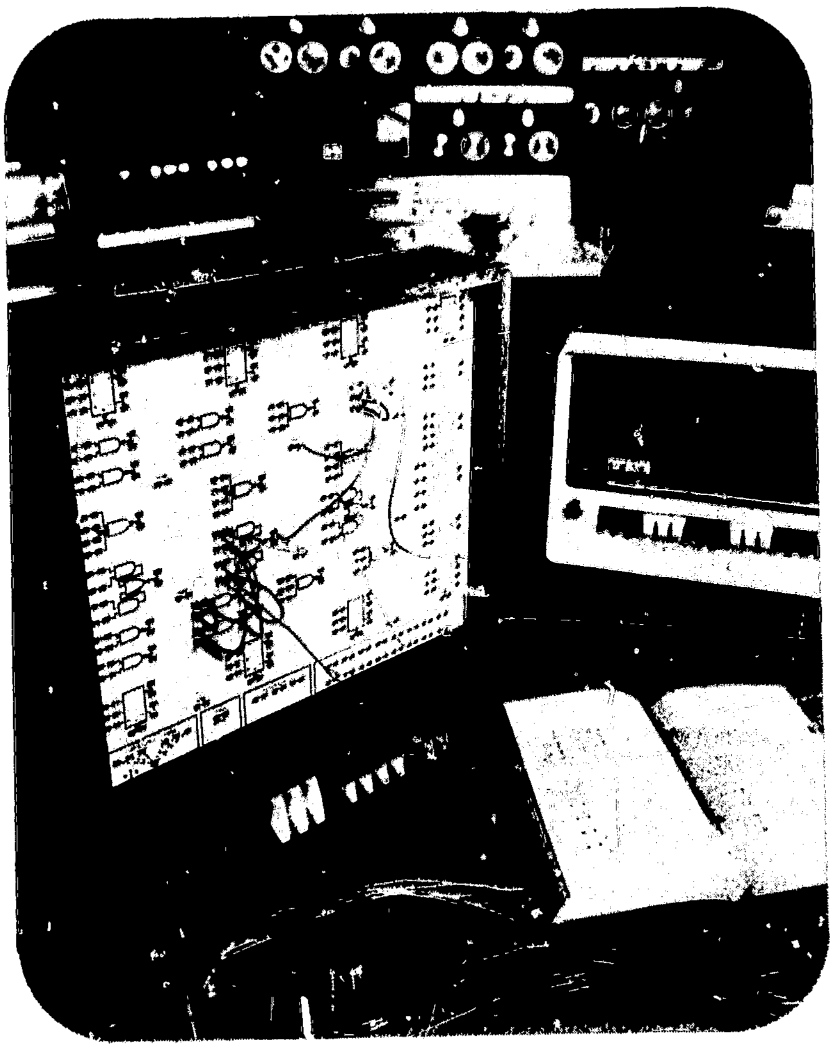
If the business curriculum is committed to computer instruction, rather than the operation of peripheral devices, it may be wise periodically to use a larger computer system which is actually being used for business data processing. If a classroom computer or terminal is available, however, the same concepts of data representation, programming, systems analysis and computer operation can be taught. The business environment would be missing, but the concepts would not change significantly.

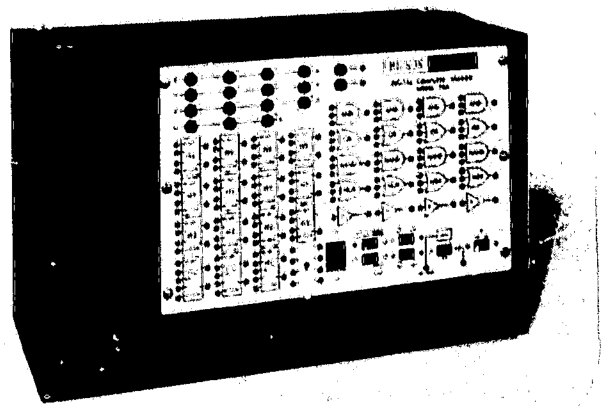
A program of prevocational training could utilize a classroom computer for practice in machine operation and programming in machine, assembly and compiler languages. In addition, vocational training in computers always should include as much experience as possible in a real world computer installation. Marc Brann, a teacher at Marconi Technical School in San Juan, California, for several years has taught a vocational course in computing for potential dropouts. His students attend the computer class all morning every day for a semester. The training includes programming and handson operation of a simple classroom computer, the ECP-18, and is supplemented by frequent visits to working computer installations in the Sacramento area. Later, students are placed with computer professionals for a period of apprenticeship. Mr. Brann has had many students with real "success stories," yet the computer he is using is in the low cost range, under $15,000.

SELECTING A COMPUTER

The Computer Instruction NETWORK used a variety of computing equipment for a variety of objectives. These included small computer logic trainers, programmable calculator, two models of portable classroom computer, a larger computer mounted permanently in a van for school visits, and the use of a time sharing service with remote terminals. Advantages and disadvantages were inherent in the use of each type.

Trainers usually are designed for instruction in computer logic, the basic circuits used in digital computers. Logic trainers, or computer trainers, can vary in price and sophistication from under $100 to several thousand dollars. If a course includes a comprehensive study of the nature of the computer and how it operates, a trainer can be a very helpful laboratory device for student experiments. Computer logic usually would be introduced early in the course, before a study of computer programming. Experiments should be kept simple for average classes unless students are training for a vocation in electronics or computer maintenance. An electronics teacher might use a trainer to include a unit on computer logic in an existing electronics class and could progress to more complex experiments.



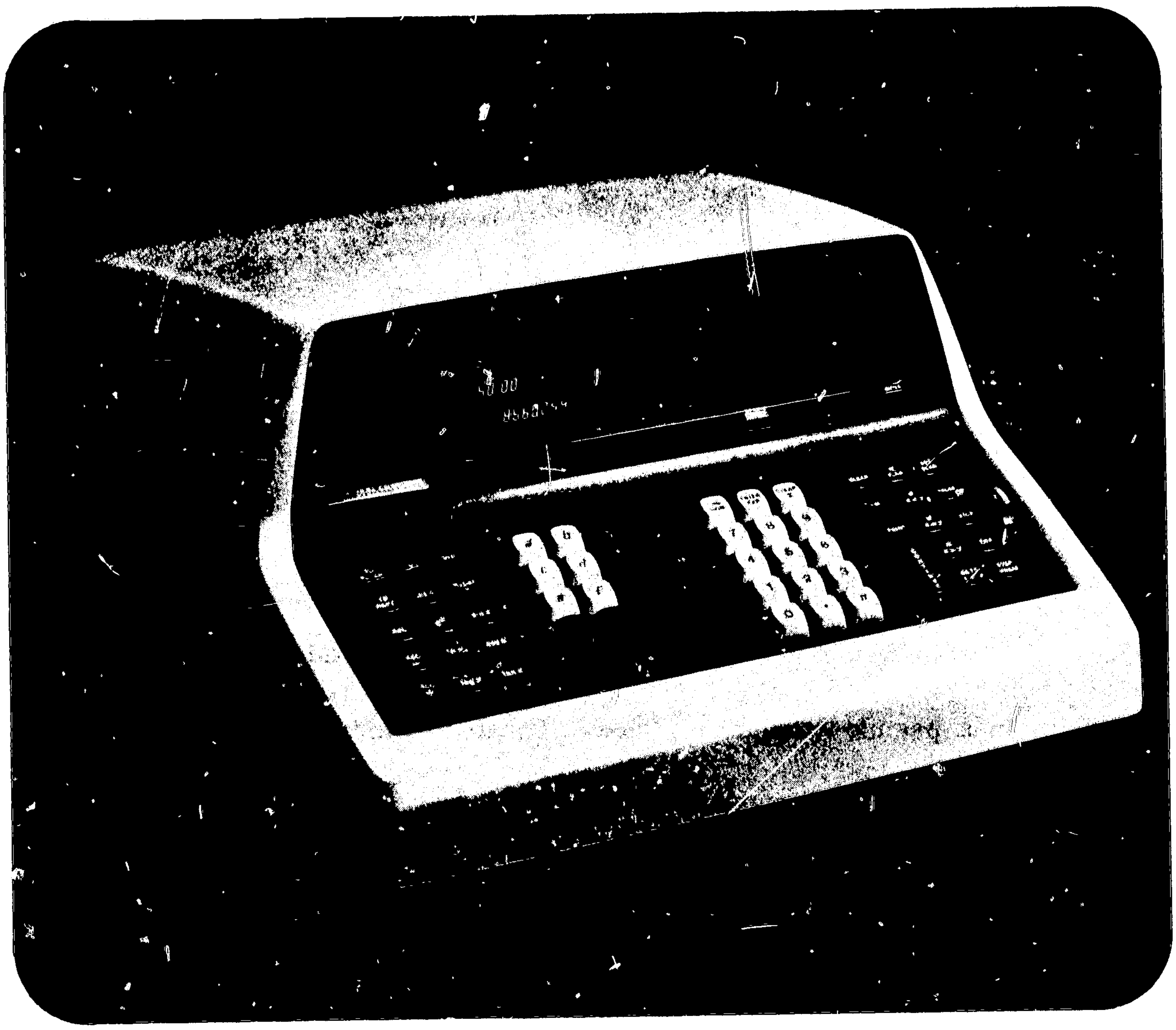


Logic trainers

Logic trainers

PROGRAMMABLE CALCULATORS

Programmable calculators are electronically similar in design and function to the ordinary desk top calculators familiar for years in many offices. However, some present day calculators have added features which place them in a pseudo-computer class. These machines, sometimes referred to as "desk top computers," are equipped with limited storage capability and may be programmed. A predetermined list of operations may be stored in the memory and executed automatically. For mathematical problem solving, calculators of this type are preferable in some cases to a real computer, as they are wired to perform automatically certain complex functions which would require many computer program steps (i.e., multiplication and division, square root, logarithmic functions, trigonometric functions and other special mathematical operations). The use of a programmable calculator for mathematical problem solving many times turns out to be less expensive than the use of a classroom computer or time sharing terminal.



A programmable calculator

Some disadvantages of the programmable calculator for computer instruction are:

The internal structure of the memory is not typical of a digital computer. That is, instructions and data must occupy certain rigidly specified areas in memory and these areas cannot be expanded in size. In most computers, the memory is not defined in this way. Instructions and data can be stored anywhere. A program may consist of 10 instructions and 100 data, or 600 instructions and no data.

A calculator cannot be programmed to modify its own instruction, as can a computer.

Calculators handle only numeric information, while computers can be programmed to recognize, interpret and print alphabetic information. This may not be a severe limitation if students only wish to do arithmetic computation and do not need to label answers.

Although calculators have a number of automatic operations which cut down on the number of steps required in programming, the memory is still limited when compared to a digital computer.

In general, then, programmable calculators can be used very effectively in mathematics classes and can even be used for limited instruction in programming. It should be emphasized, however, the organization of such a calculator is not typical of a computer and neither are the input-output methods or devices.

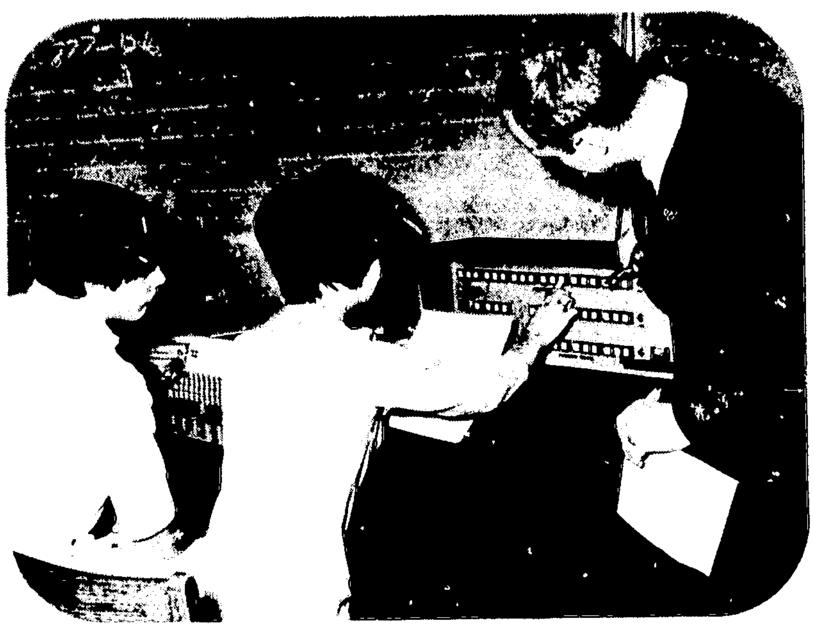
SMALL COMPUTERS

In the C. I. NETWORK computers with a small memory capacity and physical size were considered "small computers." Computers the size of a breadbox with 4096 storage locations may be scorned by some data processing professionals, but the fact remains such a computer is quite satisfactory for instruction. Since some 30 schools were to make use of the four small computers available in the NETWORK, it was imperative the machines be easily moved from school to school. Each of the four small computers (two PDP-8/S and two ECP-18) were supplied with a Teletype for input and output. These Teletypes, in addition to the keyboard, had a paper tape punch and reader. The classroom computer was available 24 hours a day--and some students arranged to keep it busy the entire 24 hours. As long as there was no limit to the hours available, students were encouraged to explore unique and challenging problems of their own invention.

Small computers do have some disadvantages. Limited memory size made it difficult or impossible to program some problems, particularly those involving large matrices. The use of Teletype input-output and handson use of a friendly computer were not typical of the equipment or operating methods found in large computer operations. The limited power of a small computer tended to limit students' understanding of the potential and diversity of application available in a large computer system.

Teachers in the NETWORK found the limitation on size of programs usually was not a significant problem. Even a matrix inversion could be performed if the size of the matrix was kept small, and the principles used in writing the program were exactly the same as if the matrix were much larger. Virtually any problem which could be solved on a large computer could be solved in a smaller simulation of the problem on a small computer. In fact, simplified illustrative examples of larger problems were preferred for educational purposes.

The PDP - 8/S computer

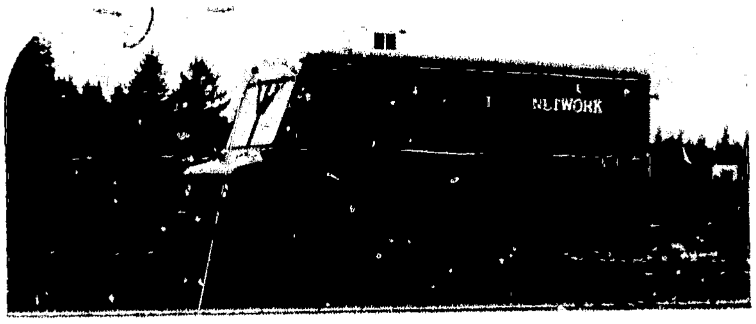


The ECP - 18 computer

THE COMPUTEMOBILE

The Computemobile was put into operation by the C. I. NETWORK as an attempt to bring a computer laboratory to the students. The mobile computer was an IBM 1130 equipped with auxiliary disk storage. The 1130 was permanently installed in a van, which travelled regularly to NETWORK schools for student use. Whereas the smaller, portable computers were left in a school for several weeks and then removed, the Computemobile visited a particular school regularly for a complete semester or year, while still serving other schools.

The first year of operation the Computemobile was assigned to visit remote, rural schools. These schools (in Lincoln and Yamhill Counties) had no access to a computer in a business or university because of their remote locations. Long distance charges for time sharing were prohibitive. They wished to offer longer courses than the NETWORK could provide, as the portable computers were to be left in a school for only a few weeks. Thus the Computemobile provided a handson computer laboratory at the school two or three times a week for the duration of the course. It soon developed that, while the computer and disk offered no unusual maintenance problems, the van itself was showing the strain of the weekly 600-mile schedule. Van driver David Gillette, an accomplished computer programmer and experienced teacher, was spending valuable hours simply truck driving. The second year of operation a schedule was developed to provide these




The computemobile goes to school


Students learn to use the IBM 1130


Class ends for the day

rural schools with a portable computer alternate months. By this time a load-and-go version of BASIC had been developed for the portable PDP-8/S computers. This meant students still had the use of a compiler language, even though they no longer were on the Computemobile schedule.

During the second year the Computemobile was tested in a new environment making short, daily trips between schools in Marion County. All of the schools visited were considered rural schools, but were much closer together than the remote schools served the first year. The Computemobile also was readily available for demonstrations, PTA talks, carnivals and short computer appreciation units.

HARDWARE SPECIFICATIONS

If the instructional objectives indicate a computer should be acquired for instructional purposes, the next step is to define its specifications. Before talking to manufacturers, the planning committee should have a fairly clear idea of the characteristics desired and should assign some priority to the specifications. For instance, is it more important that the machine language be simple and easy to learn or that the computer have a load-and-go compiler?

Although the specifications can become much more detailed, the planning committee should be able to define several basics.

Languages

All computers have a machine language and most have assembly languages. Most also will be supplied with a compiler or compilers. The name of the compiler language(s), or at least the necessary characteristics of the compiler, should be specified. One of these characteristics should limit total compile time for a single student program to a few minutes.

Speed

Any third generation computer with core memory will have more than adequate speed for instructional purposes. If the computer is a transistorized model, it probably still will be fast enough. If the memory is a drum or disk, the access time will be considerably slower than with core memory, but probably still will be within fairly reasonable limits; that is, even a second generation computer with magnetic drum memory will be slowed down by the speed of the input device.

Cost

Today, the cost of a third generation digital computer equipped with a compiler ranges from $8500 upward to infinity. A range of $10,000 to $25,000 is probably typical for a small computer. Additionally, the planning committee should insist on knowing exact costs for:

> Necessary auxiliary equipment (keypunch, Teletype, punched card reader, high speed paper tape reader, optical card reader, printer, auxiliary storage)

Installation of all equipment

Shipping and delivery charges

Maintenance contracts for all equipment, including auxiliary or "peripheral" equipment

Most small computers are equipped with Teletypes, which provide keyboard and paper tape input and output. It is desirable to have at least one extra Teletype for punching program tapes to avoid tying up the Teletype connected to the computer. A computer with a punched card reader as the input device requires one or more keypunch machines for punching student programs.

A high speed tape reader connected to a computer of this type will greatly improve throughput of student programs, as the Teletype tape reader is a comparatively slow device.

A computer smaller than 4K, although it probably would not provide a compiler, still may be valuable for basic instruction about computers; particularly if it is a simple machine and if a time sharing terminal is available for more sophisticated computing.

Larger computers have memories of 16K, 32K, 64K and upward, and the cost increases accordingly.

Other considerations which might be important to a school district acquiring a computer are:

Proximity of a maintenance facility

Power requirements

Portability

Instructional resources available

Training provided by the manufacturer

Expandibility (can the memory be increased later, and can remote terminals be added if desired)

Special environmental requirements, such as air conditioning

Number base used (base 18 is much simpler to teach than base 16, if machine language instruction is planned)

Delivery time

Once a preliminary list of specifications has been prepared, several manufacturers should be contacted and asked to send a representative. If bids must be taken, representatives of several companies should be consulted before asking for bids. They may be able to point out other considerations which should be included.

One word of caution: don't be too impressed by a "good deal." Some manufacturers are offering older machines at a substantial discount to schools. On the face of it, the discount appears to cut the price well below that of newer computers. However, the peripheral equipment, such as keypunch machines, line printer and disk drive, are not discounted at the same rate. By the time the user has assembled the equivalent of a similar but much newer system, he is paying almost the same price. If lease or purchase of a discounted system is contemplated, it would be wise to talk to someone who has used that computer to determine the equipment configuration which would meet the school's requirements. Then the manufacturer should be asked for a complete list of costs for this configuration. With the total figure in hand, compare the proposed system with other available systems at a nearly equivalent price. Weigh the cost difference against the differences in age, reliability, speed and capability.

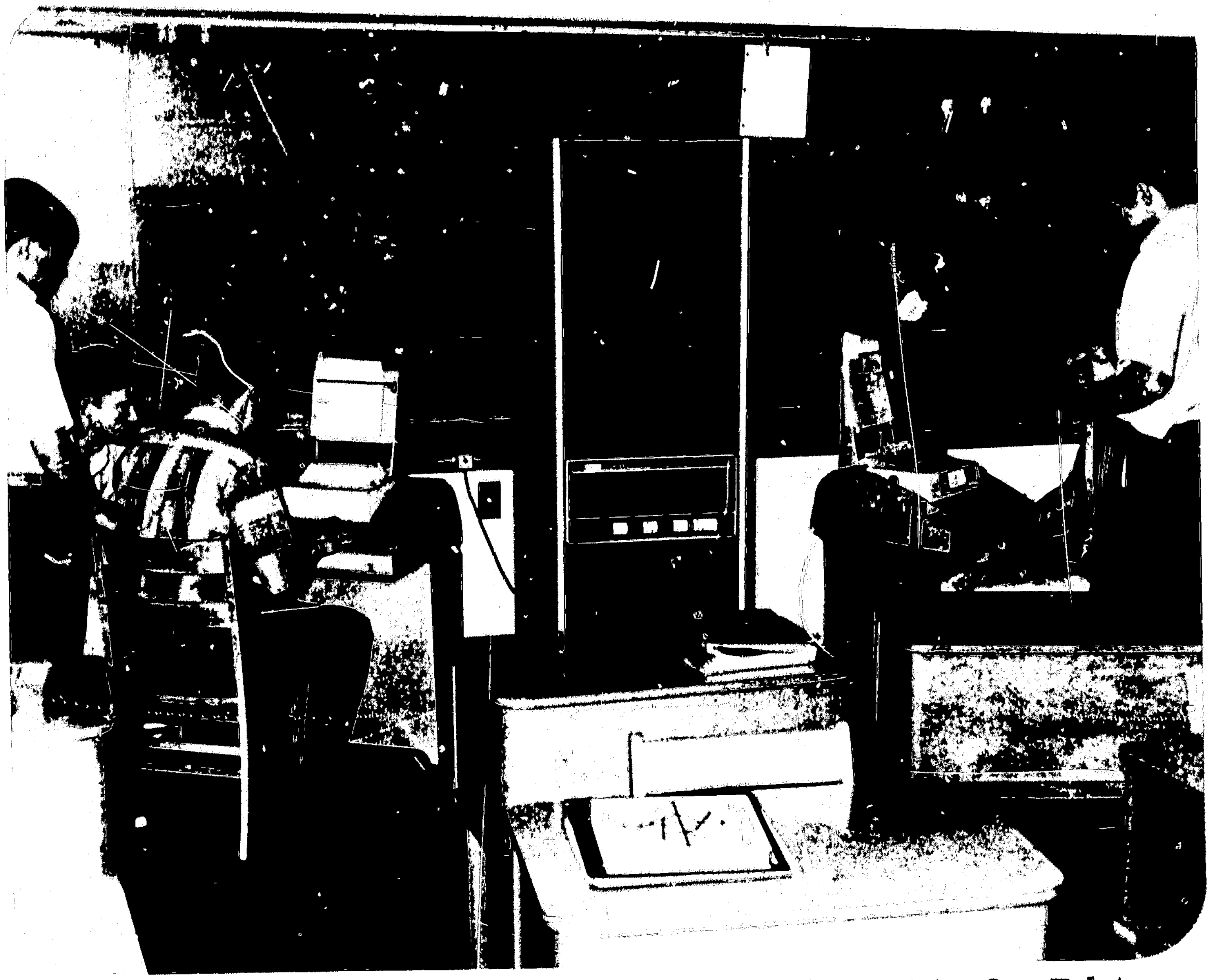
INPUT EQUIPMENT

Input equipment for classroom computers can prove to be the biggest bottleneck in getting student programs run during a class period. If, for example, a teletypewriter is supplied with the computer, students probably will use it to punch their program tapes, then load those tapes via the same teletypewriter. This means the computer will be tied up during the entire process. If an extra teletypewriter is available, program tapes can be punched offline, freeing the computer's teletypewriter. However, the speed of the teletypewriter, 10 characters per second, is relatively slow. A student program tape of 700 characters, not a very long program, will take over a minute to load into the computer. And a long program of 6,000 characters would load for 10 minutes. In a class of 30 students, this would mean that only a few students would be able to load and debug a program each hour. The added cost of a high speed tape reader would solve this problem, but students still would be required to spend time punching their own tapes.

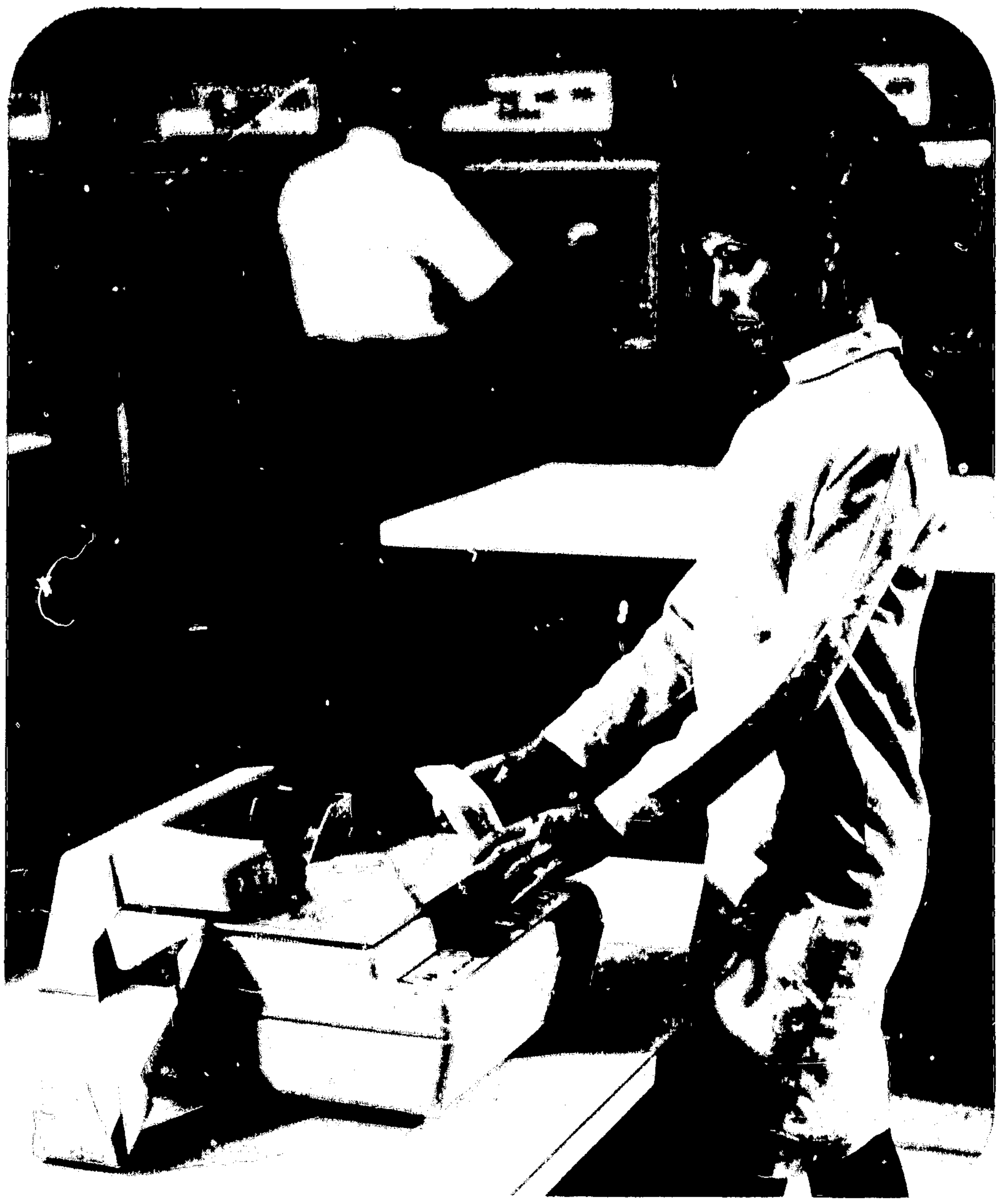
Punched card readers, reading upward from 100 cards per minute, can speed up the loading process somewhat, but require keypunch machines be available for student use. The cost of a card reader is about five times that of a teletypewriter. Students still must keypunch their programs on cards.

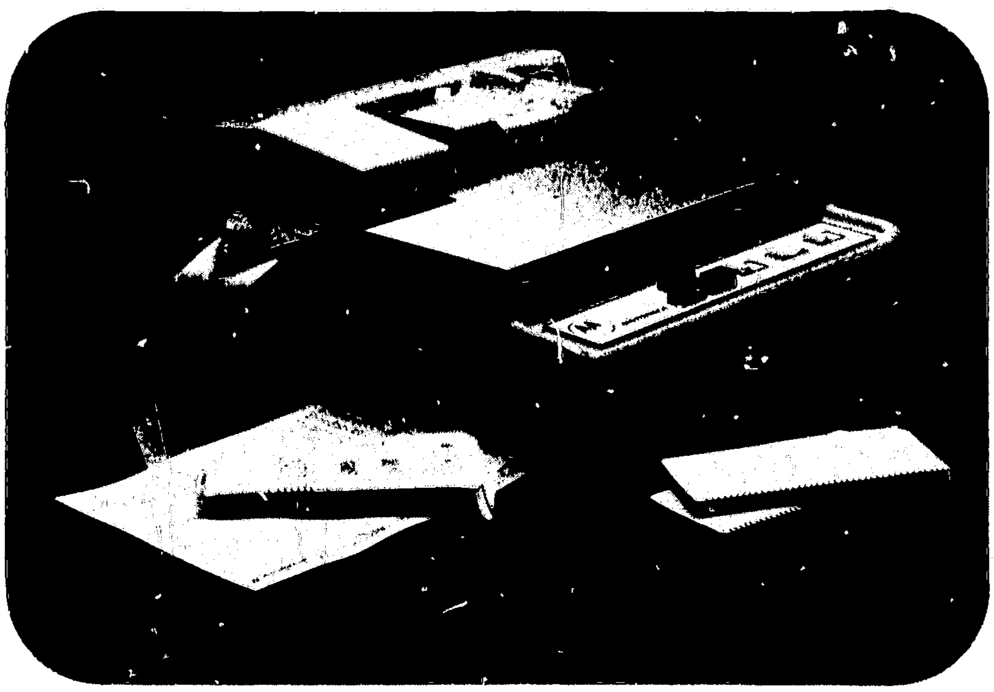
A development which offers a promising first step toward the solution of the input bottleneck is the optical mark reader. These small tabletop readers will read marks made on a standard tab card similar to those used in a punched card reader. They also will read punched cards. The speed is about 10 times as fast as a teletypewriter. Students do not have to wait for their turn at a keyboard or keypunch to punch programs, but instead simply mark their cards with an ordinary pencil.

The ultimate solution for input of student programs in a classroom has not yet been produced economically. The ideal probably would be a reader which would accept as input an 8 1/2 by 11 sheet of paper with the program simply typed.



A PDP -8/I computer installed in the Salem School District. One Teletype serves as input-output for the 8/I and one is used as a remote terminal to G. E. Time-Sharing.



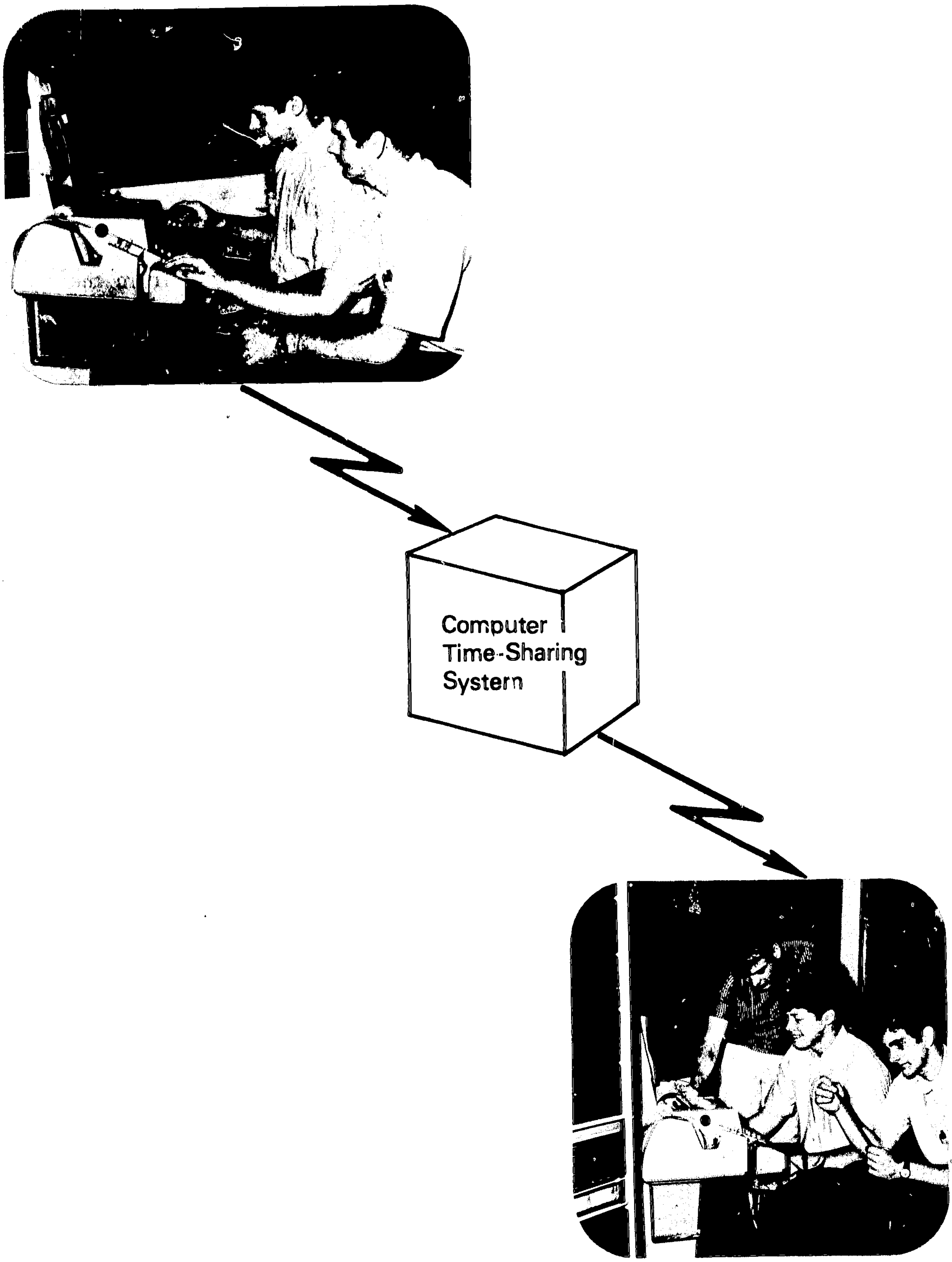


An optical mark reader, capable of reading marked or punched cards

TIME SHARING

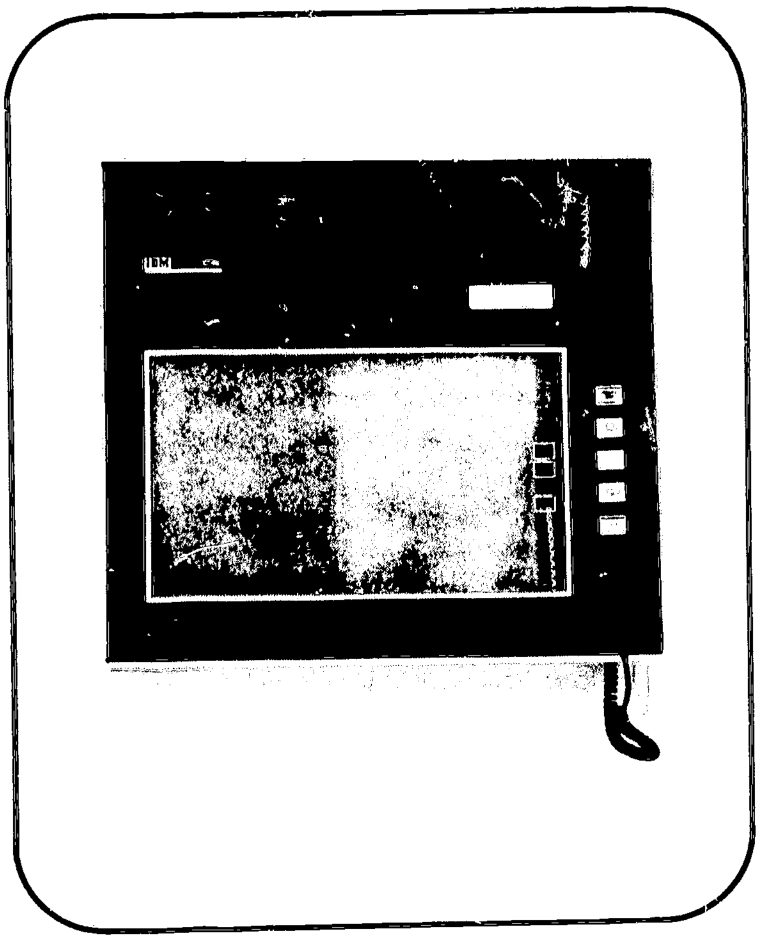
Time sharing is being used more widely by schools as commercial and university computer systems make services available. In such a system, a single, large, fast computer is able to accept input from dozens of remotely located terminals, all at the same time. These computers can operate in billionths or trillionths of a second, so are constantly "input-output bound." That is, human typists and mechanical input devices with their relatively slower speeds keep the computer waiting much of the time. Thus, it is possible, indeed practical, for a high speed computer to handle the input processing and output for many terminals at once, with no discernible lag in time for the user sitting at a terminal 25 miles away from the computer. To the user, it appears that the computer is dedicated to handling his unique problem.

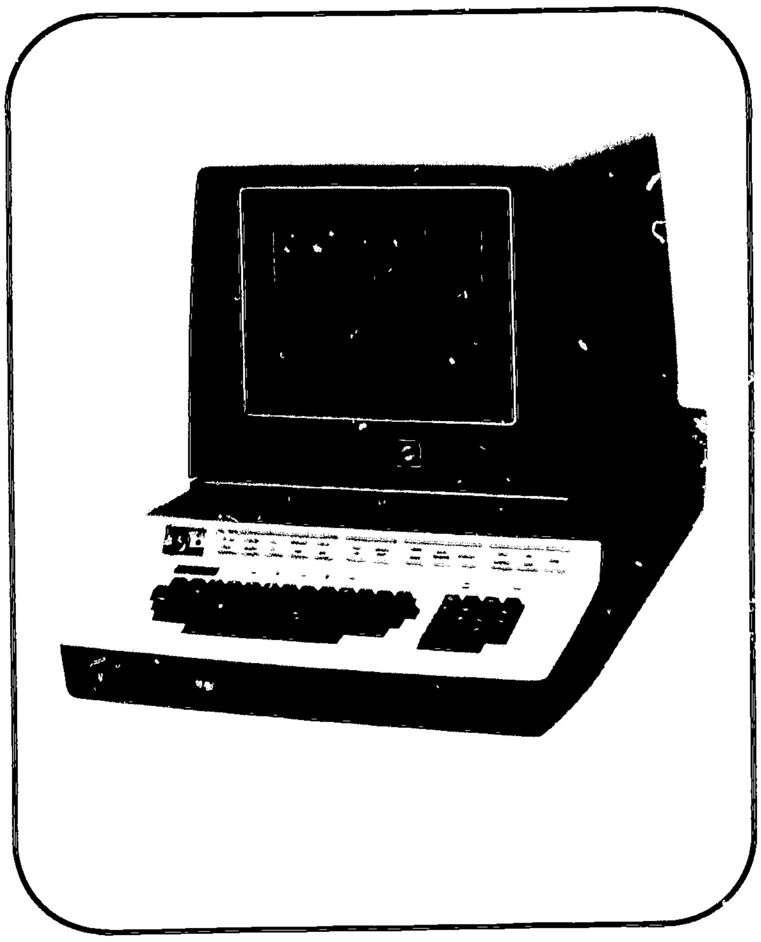
A variation on the time sharing scheme, and a more inexpensive way to use a remote computer, allows the school user to enter a program or, more often, a series of programs, from the terminal. These programs then are recorded on an offline storage device, usually magnetic tape, and processed in a "batch" at a slow time, usually after midnight. The resulting output, then, is printed out at the school at a later time, often the next day. Alternatively, the output is run off on a high speed printer at the computer center and mailed to the school.

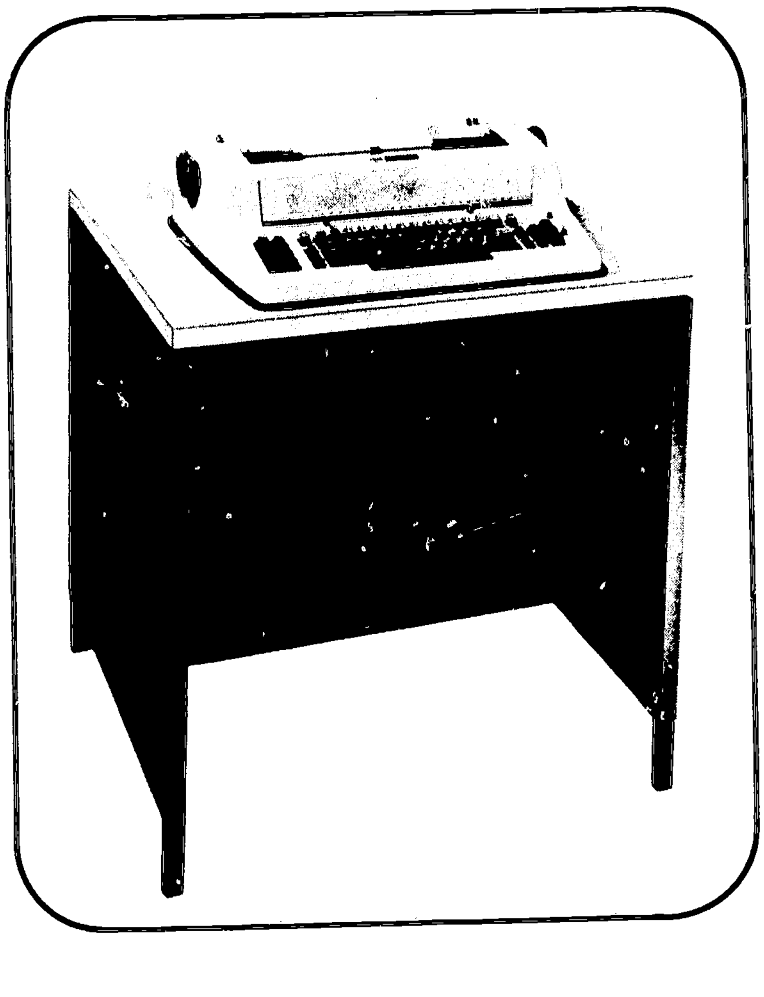

Computer
Time-Sharing
System

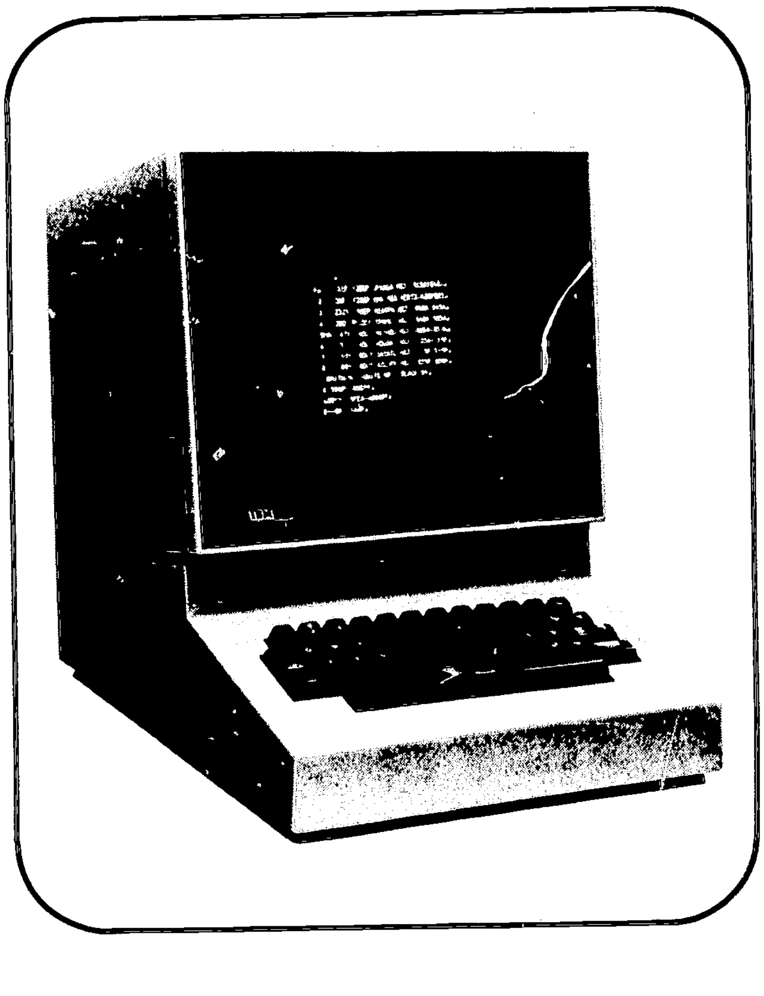
The units and courses established in most NETWORK schools provided for an introduction to computers with programming in machine language and handson operation of a portable computer. This introduction was followed by the use of a high level language--in most cases, BASIC--for problem solving at a remote terminal. The terminal, a standard ASR 33 Teletype, was connected to a telephone line and leased on a monthly basis from the telephone company. The simplicity and power of the BASIC language allowed students to solve complex problems with relative ease, without worrying about storage requirements or compile time. The full capabilities of a large, modern day computer system thus could be made available to students and teachers in a corner of the classroom.

If computer appreciation is an objective of the instructional program, then the exclusive use of a time shared terminal is a drawback. Students' misconceptions are likely to be reinforced by the enigmatic, seemingly omnipotent terminal. The remote computer is a vaguely comprehended device, and the language appears to be one which the computer mysteriously but immediately "understands." Only the study and use of a simple computer, even if it is a model computer developed on the blackboard, helps dispel the mythology of the computer.









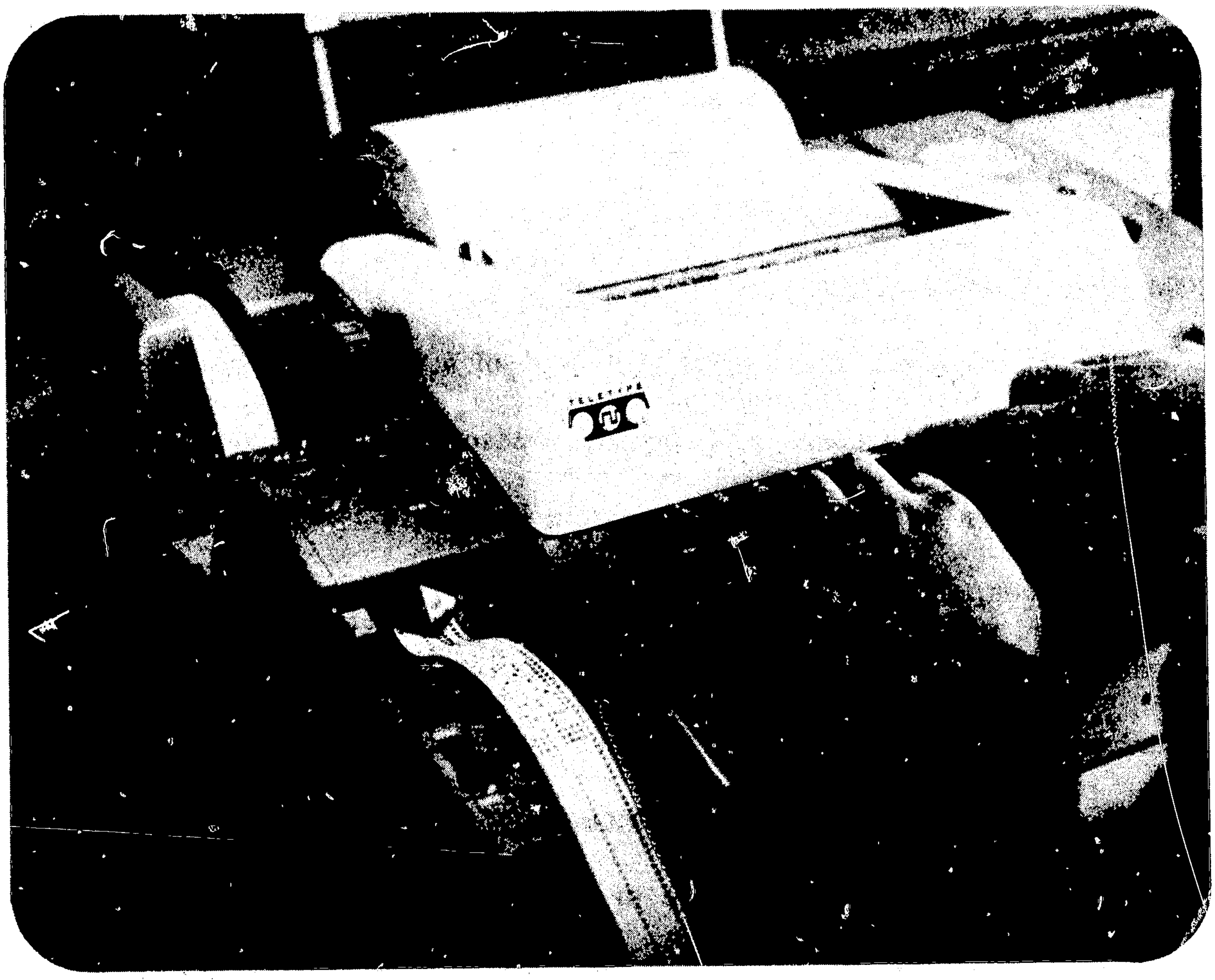
Another drawback in using time sharing can be the cost. This cost is coming down daily, as more and more commercial systems increase competition and improve the quality of time sharing services. However, the 24-hour-a-day availability of a classroom computer at no increase in cost might be a preferred alternative to the current cost of $10-$20 per hour for use of a time shared system.

A wise approach to the use of programmable calculators, small computers, and time shared terminals is one suggested by Robert Albrecht, Senior Consultant at Portola Institute in Menlo Park, California. All three types of hardware would be available in the school, and students would be issued "scrip" at the beginning of the computer course. They would be expected to pay for the use of any piece of hardware at predetermined rates, using this phony money. Lowest rates would be for use of the calculator, highest rates for the terminal. This system would require the students to analyze their problems to decide on the most economical tool to be used. In some cases, a slide rule or pencil and paper might be the least expensive!

TERMINALS

Terminals available on time sharing systems can include keyboards, cathode ray tubes and plotters. While most systems will handle only keyboards, punched paper tape and/or punched cards can be used as input media with some terminals. The use of paper tape or cards can greatly reduce terminal time and resultant cost, since the device can be operated for input at speeds a human operator could not sustain. Terminals also can vary in their input and output speeds and in their ability to handle upper and lower case. For most instructional uses the extra cost for the upper-lower case feature cannot be justified.

The teletypewriter probably is the most used terminal for instructional applications, and it also is the least expensive. The Teletype Model ASR 33 is equipped with paper tape punch and reader, and can be purchased from



A Teletype Model ASR33

Teletype Corporation or leased from the telephone company. However, any terminal used for time sharing must be connected to the computer by a communications line. If the teletypewriter is leased from the telephone company, the telephone line is a part of the monthly lease cost. If, however, the teletypewriter has been purchased, it must be supplied with an acoustic coupler which will allow the use of any ordinary telephone to connect the teletypewriter to the time shared computer. Some time sharing services provide for leasing a portable teletypewriter with an acoustic coupler. This unit can be carried from room to room and used in any location where there is a telephone.

COMMUNICATIONS LINES

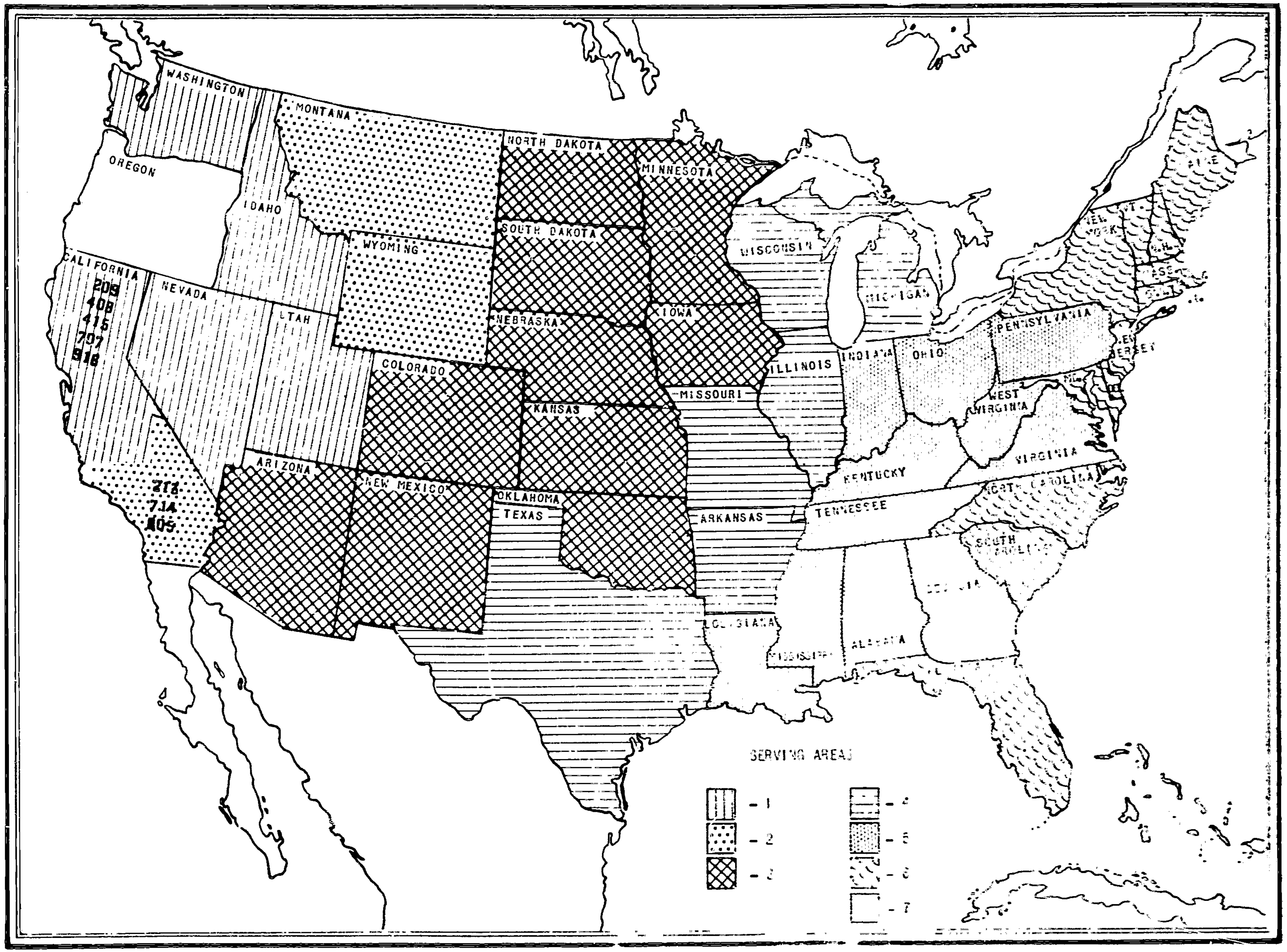
Communications lines are a vital part of a time sharing system. Without them, remote users could not be connected online to the computer. Two broad categories of service are available from a telephone company--dialup services, which are similar in operation to a home or business telephone, and private line services. Private line services are lines made available for the exclusive use of the customer and are available at all times for use as required by that customer.

Three services are available in the dialup category. The first is DATA-PHONE service, which is a fairly recent offering of the Bell System. This service uses the regular telephone network alternately for data and voice transmission purposes. A unit of equipment known as a data set converts the electrical impulses generated by the terminal device into tones which are suitable for transmission, and reconverts the tones back into the original impulses at the receiving end. DATA-PHONE sets are compatible with a number of business machines and terminals, including several models of teletypewriter. The data sets can transmit messages at speeds from 75 B.P.S. to 2000 B.P.S. (Bits Per Second--a bit being a single binary character). The slowest transmission speed is approximately equivalent to the speed of a Model 33 Teletype, about 100 words per minute.

The second dialup service used by the NETWORK was Teletypewriter Exchange or TWX. Approximately 60,000 teletypewriters are presently using TWX service. TWX transmission speed is about 100 words per minute, or 10 characters per second. A faster line is not necessary, as 100 words per minute is the maximum speed of the teletypewriter.

Wide Area Telephone Service (WATS) is the third type of dialup service available. Calls are made on the regular long distance network, but are limited to a selected area of the country. This service can be outward (you can call out but no one can call you) or inward (you can receive but not place calls), and is charged at a set monthly rate rather than by individual call.

WIDE AREA TELEPHONE SERVICE



If a school district is using a time shared computer in the same city, DATA-PHONE service is the best of the three. No charge is made for local calls, other than the regular monthly rate. TWX, on the other hand, charges the monthly rate plus a charge for every local and long distance call. WATS is practical only when a large volume of long distance calls are made regularly.

The cost of time sharing will increase if a district must make long distance calls to reach the computer. If this is the case, any of the three dialup services would suffice. However, the monthly charge for a WATS line is considerably more than the basic rate for DATA-PHONE or TWX and would be practical only when the number and length of calls exceeded the charge for an outward WATS line.

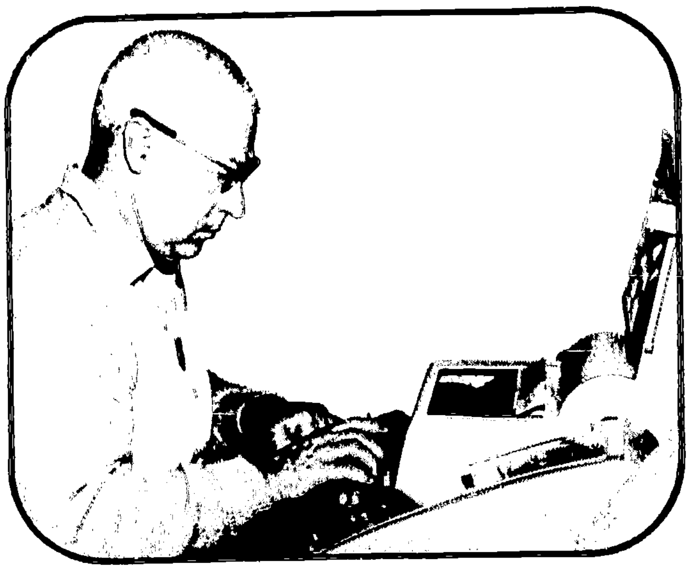
Private line services are available at a higher cost than dialup services and transmit at a much higher speed, higher in fact than is required for most school time sharing applications.

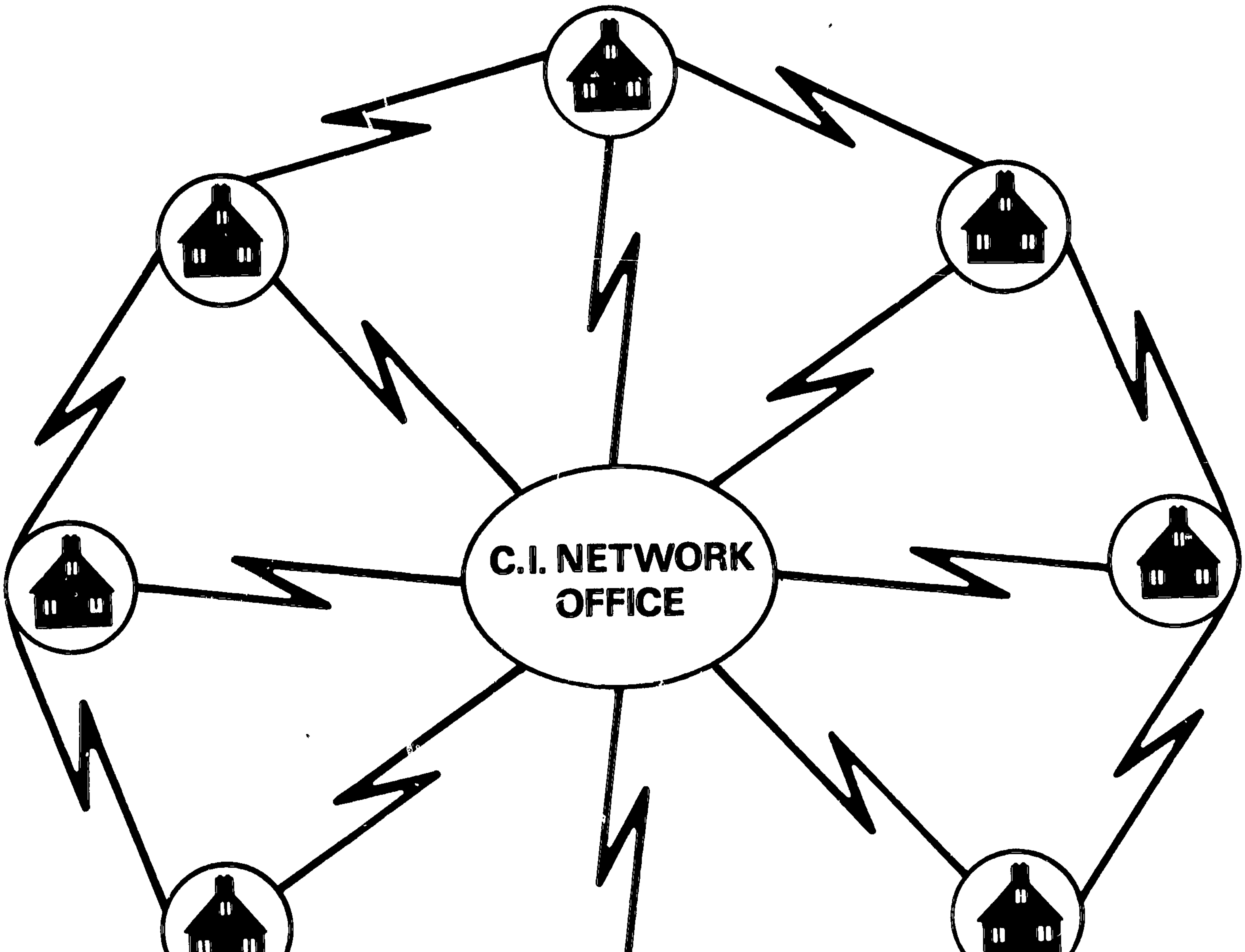
THE NETWORK CONCEPT

In the Computer Instruction NETWORK, time sharing was only one of the many uses for a standard teletypewriter. Since each of the four portable computers had a teletypewriter with keyboard and paper tape input-output, the teletypewriter terminals installed for time sharing also could be used to punch paper tapes for the portable computers. The extra teletypewriter effectively doubled the throughput on these smaller machines, as a student using the computer would usually prevent the computer's own Teletype from being used to punch paper tapes.

Since the IBM 1130 installed in the Computemobile was equipped with paper tape input and output, the teletypewriters in the schools could be used to prepare paper tapes prior to the arrival of the van. This o 'ine tape punching capability is even more essential for a mobile computer tnan for a computer which remains in the classroom.

A particular advantage of having teletypewriters and telephone lines installed in participating NETWORK schools was the communications capability provided by TWX service. Not only was every school connected to every other school via telephone line, but each teacher could reach the NETWORK office by simply dialing the number on his teletypewriter. During the first uncertain year, this proved to be a remarkable source of aid and comfort for teachers, and a method for the NETWORK staff to send important messages to participants. Some days the teletypewriter in the office would "turn on" every few minutes with a question or an urgent plea for help from a teacher. Often, the question could be answered quickly and easily. Sometimes a maintenance man would be dispatched. Other times a staff member would go to the school to work with the teacher or solve a computer or curriculum problem. This means of instant communication probably made the difference between success and failure in the first crucial year. Too often in a pilot program the teacher is trained, equipped and given a friendly shove, never to be heard from again. In computer instruction, perhaps more than




C.I. NETWORK
OFFICE

anything else, the novice teacher needs continuing consultant help, assistance and assurance from experienced and qualified people. The communication line brings such assistance directly into the classroom at the moment when the problem is encountered.

Students soon discovered a fourth use for the teletypewriter and TWX lines. When two schools were alternating use of the same portable computer, a month at a time in each school, students organized a "remote programming" pool. As soon as the computer would be transferred to school B, students in school A would begin to transmit programs to volunteers in school B, via the teletypewriter paper tape reader and telephone line. The school B volunteers would try the program on the computer and type results (or comments about the absence of results) back to the original student. Those participating in this sharing program soon developed better skills and techniques for communicating.

COSTS OF COMPUTER INSTRUCTION

If a short unit in computer appreciation is incorporated into an existing class, the cost for teacher time, materials and equipment is negligible. If, however, a unit or course(s) of any length is established, certain costs can be expected for teacher training, released time for teachers to plan the course and develop materials, resource and library materials, student texts, computing equipment, communications costs, supplies for the computer and maintenance.

As in the development of any pilot course, teachers should be trained and given time to plan and prepare preferably during the summer months. During the first year of operation, the teachers involved also should be given released time to continue developing materials and testing computer programs.

The usual costs for resource books, periodicals, films and student texts should be expected for a computer course. If a classroom computer or terminal is to be used, the manufacturer can quote an expected budget figure for supplies such as Teletype paper, paper tape, punched cards and programming forms. The major part of the cost, however, will be in the lease or purchase of equipment.

TRAINERS

Training devices can be found which sell for as little as $4.95. Some of the inexpensive, simple trainers do little more than show the operation of a binary "flip flop," and some must be self-assembled. Some are made of flimsy plastic and seldom operate after the first feeble attempt. Since a trainer might be designed to teach digital logic or number systems, or simulate completely all of the actual functions of a digital computer, the price can range upwards to as high as $6,000. Some small computers are called trainers by those manufacturers who are selling a larger computer. Usually, if the price is over $6,000 and the word "trainer" does not appear in its name, the device can be considered to be a computer. This is rather a superfluous test, but is a fast way to make an initial distinction. The average price for a computer trainer is $150 to $500. A digital logic trainer might sell for as little as $140 or as much as $1500. Students in an electronics class might be persuaded to build a logic trainer for much less, if they were provided with specifications and the necessary components.

PROGRAMMABLE CALCULATORS

The price of a programmable calculator depends upon the number and sophistication of its builtin arithmetic functions, the sophistication of the input and output, and the size and nature of its memory. Every major manufacturer of calculators has announced a calculator "with a memory."

Purchase prices range from about $1,250 to $5,000, and options can include such sophistications as remote control keyboards for "multiple station applications," an optical mark reader or punched card reader for input, printout or cathode ray tube display for output, expandable memory and hard wired or programmed subroutines.

SMALL COMPUTERS

As the miniaturization of computer circuits increases the speed of computers, the physical size and cost decrease. The use of "integrated circuits" makes it possible to cut both manufacturing and maintenance costs. A typical example of the recent trend to more compact, faster, low cost computers is the PDP-8 line manufactured by Digital Equipment Corporation. Two years ago the PDP-8, a relatively small but fast computer, was selling for about $18,000. Soon a little brother, the PDP-8/S was introduced, 13 times slower but $8,000 cheaper at $10,250. Last year, DEC impacted their own PDP-8 and 8/S market by announcing the PDP-8/I and the PDP-8/L, each the same speed as the original PDP-8 (13 times faster than the 8/S), each using integrated circuits, and selling for $12,500 and $8,700, respectively. Although both the 8/I and the 8/L have a basic 4096-word memory, the 8/I can be expanded and even used with multiple terminals. This decrease in cost, while increasing the speed and reliability of the machine, is typical of the trend. However, this should not prevent educators from committing themselves to a particular machine for fear it will be replaced by one better or faster. The NETWORK's PDP-8/S computers have not been obsoleted and are still as useable and appropriate as ever.

A real computer with at least 4096 words of storage (or "4K") can start at less than $5,000. In fact, a 4K computer with teletypewriter input-output can cost from $6,000 to $25,000, depending on the manufacturer and factors such as speed, expandability and capacity of the individual storage cells. For instance, a "word," or a single storage cell, can vary from 8 binary digits in length up to 24 or 32 binary digits. It would be wise to check the true storage capacity of a "4K" computer. If the word length is only 8 binary digits the true capacity is about half that of most 4K computers.

If a school district is serious about building a curriculum in computers and using computers to extend the curriculum in many other areas, the computer acquired should be expandable. That is, the district should be able to add additional memory and additional remote keyboards at a later time to permit more students in more classrooms to make use of the single computer. Alternatively, if the computer is not to be expanded, remote terminals can connect students to a commercial time sharing service or university.

Expanding the memory and remote keyboard capability of a machine will naturally expand the cost. The cost of a basic 4K computer with a single keyboard might be between $10,000 and $15,000 while the same system equipped to handle eight remote keyboards would cost at least $50,000 not counting the cost of the eight additional keyboard units.

TELETYPEWRITERS

Model 33 Teletypes with paper tape attachments can be purchased for about $675 from the Teletype Corporation. If the teletypewriter is equipped with an acoustic coupler, the cost will be about $600 more, which is the average cost of a coupler. If, in addition, the teletypewriter is attached to a cart on wheels, the entire device would sell for about $2,000. This $2,000 teletypewriter, then, could be used with any ordinary telephone to dial a time sharing computer system. Many time sharing services will lease this portable terminal for about $125 a month.

If a school district purchases teletypewriters to use for offline tape punching, budgetary provision should be made for maintenance. Since the Teletype Corporation at this time does not have a service organization, it would be up to the district to find a person qualified to do teletypewriter repairs. The average monthly cost for such maintenance is difficult to predict, as some teletypewriters (like some automobiles) may run relatively maintenance free for thousands of hours of use, while others may need to be replaced after 100 hours of use. In the NETWORK, two teletypewriters were purchased. When heavily used the monthly cost for maintenance over a year of operation averaged about $15 each.

While the telephone company will lease Teletypes and provide all maintenance, they are prohibited by the FCC from leasing a Teletype without a telephone line, simply as a tape punching device.

TIME SHARING

Determining the true cost of using a time sharing service requires determination and persistence. If only one commercial time sharing service is available, count your blessings. If there are several, a total approximate cost figure for one hour of online use of the system should be obtained from each vendor. A separate "connect time" charge may be made for the time the terminal is connected online to the computer system via telephone line. In addition, there probably will be a charge for "CPU time," or the time that the central processing unit of the computer is actually engaged in processing of the user's program. Other charges may be made for the amount of storage used in the operation of the user's program and the amount of storage used for "saving" programs or data to permit a customer to call them out again at a later time. A number of other special charges may exist. For example, some systems have an established order of priorities. By paying a little more, a customer can be assured of having his programs attended to sooner than other users waiting at terminals. Most systems have a minimum monthly contract charge, and some will provide unlimited use for a fixed monthly charge.

If costs of various services do not need to be compared, a salesman can provide an average total cost per hour of connect time, which will include CPU, storage and any necessary options, The ratio of CPU time to connect time will be lower for students than for most other users, as students will tend to write shorter programs, at least in the beginning. Some programs will run in "zero second," as the CPU time is computed to the nearest fraction of a second.

Although the cost for time sharing services is decreasing daily, an hourly cost today probably would be between $8 and $12 for instructional use. It is unlikely there will be much need for additional or permanent storage, so the major cost will be for connect time and CPU time.

In addition to the hourly cost, add costs for initiation of service--in some cases $100. If a terminal is to be leased from the company, include that cost. All of these costs will be payable to the time sharing service.

The costs for communications lines and probably terminals must be added to the time sharing costs. These charges will be paid to the telephone company. Even if the terminal is to be purchased or leased from the time sharing service, a communications line still must be acquired. The only exception would be if a portable terminal equipped with acoustic coupler is used with an existing telephone.

COMMUNICATIONS

The dialup services mentioned in the equipment section are billed as indicated on the next page.

DATA-PHONE service

Data set	$25.00 a month	$25 installation
ASR 33 Teletype	$40.00 a month	$25 installation
Telephone line	$13.50 a month	$10 installation
Total monthly charge	$78.50	$60 installation (a one-time charge)

Long distance calls are billed at the usual rate.

TWX service

Basic service	$45.00 a month	
ASR 33 Teletype	$15.00 a month	
Total monthly charge	$60.00	$50 installation (a one-time charge)

Long distance calls are billed at the usual rate, and there is a charge for local calls as well: about 15¢ for five minutes.

WATS

Interstate WATS

The country is divided into six "zones" for WATS billing purposes. The telephone company will provide information on the states included in each zone.

The rates given are for interstate service only. Intrastate WATS is governed by local Public Utility Commissions, and the rates may vary.

Rates shown are the same whether the service is "outward" (calls may be made but not received) or "inward" (calls may be received only).

Unlimited service allows an unlimited number of calls to be made each month within the zone charge. Measured service limits the calls to 10 hours per month per line. All use exceeding the 10-hour limit is billed at the hourly rate shown.

	Unlimited service	Measured service	"Extra hours" charge
Zone 1	$ 900/month	$220/month	$16.50/hour
Zones 1-2	$1300	$260	$20.50
Zones 1-3	$1750	$310	$23.50
Zones 1-4	$1950	$335	$25.50
Zones 1-5	$2200	$360	$28.00
Zones 1-6	$2275	$375	$29.50

<u>Intrastate WATS</u>

In Oregon, intrastate WATS is $575 a month. Measured WATS (15 hours per month) is $210 a month, with a $12 charge for each additional hour. If the time shared computer being used is in the same state, but cannot be reached by making a local call, intrastate WATS may be more economical than paying for each long distance call. Salem schools in the NETWORK placed calls to Portland for time sharing at about $6 per hour for long distance charges. At this rate close to 100 online hours per month would be the crossover point at which WATS would become less expensive than DATA-PHONE or TWX service. This would be between four and five online hours each school day, considerably more than the schools were using.

The installation charge for a WATS line is $10. The above rates are for the line only and do not include lease of a Teletype. The

Teletype lease would be approximately $65 per month additional, although this rate may vary from state to state.

Private Lines

Private lines are dedicated, point-to-point lines. They are billed (in Oregon) at about $3 per mile per month, plus about $10 per month connection charge at each end of the line.

REFERENCES

INTRODUCTORY

Allen, Paul III. Exploring the Computer. Reading, Mass.: Addison-Wesley Publishing Co., 1967.

> A programmed text giving general information about computers, their makeup and applications. A good review of this book by a sixth-grade child was published in Datamation, which attests to its readability.

American Federation of Information Processing Societies. The Quiet Revolution: Computers Come of Age. New York, N.Y.: The Federation.

Barnett, Leo and Lou Ellen Davis. Careers in Computer Programming. New York, N.Y.: Henry Z. Walck, Inc., 1967.

Campbell, Robert. "How the Computer Gets the Answer." Life. Reprint No. 33, 1967.

> A primer for the layman on computer components, organization and operation, with some comments on man-machine interface. Contains excellent photographic illustrations and employs an interesting "model train" to show how an addition problem is handled by a computer. Can be ordered from Life Educational Reprints for a fee.

Charp, Sylvia. Computers in General Education. Harrisburg, Pa.: Pennsylvania Department of Public Instruction, 1967.

> An introductory pamphlet which defines computers, discusses their development and introduces flowcharting and coding. It contains several good flowchart examples.

Darnowski, Vincent S. Computers--Theory and Uses. Washington, D.C.: National Science Teachers Association, 1964.

> As a general information unit for junior high and older students, the booklet discusses the development, components, operation and use of computers. It contains a section on programming and a discussion comparing analog and digital computers. Self-tests, questions and activities follow each section. Classroom sets are available. The Teachers' Guide is a useful book whether you use

the class text or not. More than half the book is an annotated list of books, films, games and articles, while the rest contains suggestions for each chapter and a test.

Darnowski, Vincent S. A Teacher's Guide to Computers--Theory and Uses. Washington, D.C.: National Science Teachers Association, 1964.

Desmonde, William H. Computers and their Uses. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1964.

A general introductory book, including information on most phases of data processing. It is mostly non-technical and readable by a non-science oriented person. Also included are two chapters on the social roles and implications of computers. Of interest to teachers might be the sets of review questions at the end of the book, one for each chapter.

Fahnestock, James D. Computers and How They Work. South Brunswick, N.J.: A. S. Barnes and Co., Inc., 1966.

An older reference, but still good for those items included. The book is relatively easy to read and much of it can be read by a non-science oriented reader. This book has many useful illustrations. No information is given about compiler languages, but it does have chapters on analog computers, basic logic circuits and memories.

Gielow, Fred C., Jr. Introducing--The Computer. White Plains, N.Y.: IBM Corporation.

Hollingdale, S. H. and G. C. Toothill. Electronic Computers. Baltimore, Md.: Penguin Publishers, 1965.

Laver, F. J. M. Introducing Computers. New York, N.Y.: British Information Service, 1965.

Los Alamos Scientific Laboratory. Automatic Digital Computers. Los Alamos, New Mexico: University of California, 1965.

McCracken, Daniel D. Should You Go into Electronic Computer Programming? New York, N.Y.: New York Life Insurance Company.

National Cash Register Company. Electronic Data Processing: Written for the Layman. Dayton, Ohio: The Company.

Three pamphlets introducing data processing from a business viewpoint. Book I, "What is Data Processing?" sketches all phases of E.D.P.

Book II, "What is Binary Arithmetic?" includes subtraction by complementary addition and examples of computer memory. Book III, "What Is a Computer?" discusses hardware. They can be used separately. Classroom sets available or can be obtained through local NCR representatives.

Programming: Words that Move Machines. White Plains, N.Y.: IBM Corporation.

Say Something in Computerese. Minneapolis, Minn.: Honeywell, Inc., 1965.

You and the Computer. New York, N.Y.: General Electric Company, 1965.

A short pamphlet concerning computer operation, programming, I/O devices and applications, suitable for use as an introduction or overview. Contains some good illustrations. Available free from GE.

COMPUTER APPLICATIONS AND SOCIAL IMPLICATIONS

Amber, George H. and Paul S. Amber. Anatomy of Automation. Englewood Cliffs, N. J.: Prentice-Hall, Inc., 1964.

Arnold, Pauline and Percival White. The Automation Age. Pound Ridge, N.Y.: Holiday House, 1963.

A general introduction to automation, readable by junior high students. It contains good sections on applications in various areas of endeavor and the social effects, especially unemployment. Information current to 1963.

Association for Supervision and Curriculum Development. Humanizing Education: The Person in the Process. Washington, D.C.: The Association, a Department of the National Education Association, 1967.

"Automation and the Decline and Fall of Work." Senior Scholastic. 91:6, October 26, 1967.

Bluemle, Andrew. Automation. New York, N.Y.: World Publishing Company, 1963.

This short, easily read book gives an introduction to automation and its social challenge. It can be read easily by junior high students. The illustrations are good; and information is current to 1963.

Bowen, Charles R. and Swayne Orton. Automation and Your Future. Nashville, Tenn.: Methodist Board of Christian Social Concerns.

Bushnell, Don D. and Dwight W. Allen. The Computer in American Education. New York, N.Y.: John Wiley and Sons, Inc., 1967.

A collection of papers presented at an AEDS conference in November, 1965, at Stanford University. The book's four major divisions deal with (1) individualized instruction and social goals, (2) computers in instruction and research, (3) teaching the computer sciences and (4) information processing for education systems. It is assumed that readers have a working introduction to computers and their application to education. The book is a summary on the state of the art of such applications with the last chapter listing the conference recommendations. Included are comments on needed changes in pedagogy and administration for effective use of computers.

Caffrey, John and Charles Mossman. Computers on Campus. Washington, D.C.: American Council of Education, 1967.

Cherry, Colin. On Human Communication. Cambridge, Mass.: MIT Press, 1966.

Computer-Assisted Instruction Guide. Newburyport, Mass.: ENTELEK Inc., 1968.

Cordell, Christobel M. The Computer in Your Life. Portland, Maine: J. Waston Walch, 1967.

Evans, George W. II., Graham F. Wallace and George L. Sutherland. Simulation Using Digital Computers. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1967.

This book is concerned with the simulation of systems by computer simulation models. It defines and introduces the concept of simulation, discusses computers and computing systems and their use as simulation tools, and presents examples of computer simulation in military medical evacuation, armed combat, war games and radar observation of space vehicles.

Feigenbaum, E. and J. Feldman. Computers and Thought. New York, N.Y.: McGraw-Hill, 1963.

Fink, Donald G. Computers and the Human Mind. New York, N.Y.: Doubleday and Co., Inc.

The two objectives of this book are to explain how computers work and to what extent computers are intelligent. It contains chapters on math, languages, flowcharts, programming, machine workings, brain-computer analogies and machine creativity. A good informational book for bright science oriented readers.

Gerard, Ralph W., editor. Computers and Education. New York, N.Y.: McGraw-Hill Book Co., 1967.

The format of this book is the same as that of the conference on which it reports. Each section begins with an address, then topics for discussion are suggested, after which the discussion takes place, followed by a summary. Three general areas are covered: CAI, library and administration. The library reference is one of few such sources this reviewer has found. The discussion sections make cumbersome reading. The report is oriented to higher education but has implications for other levels as well.

Glaser, Robert, editor. Teaching Machines and Programmed Learning, II. Washington, D.C.: Department of Audiovisual Instruction, National Education Association of the United States, 1965.

Guetzkow, Harold. Simulation in International Relations. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1963.

"The Information Explosion: The Computer in Society." The General Electric Forum, Winter 1967-68.

Kahn, Herman and Anthony J. Wiener. The Year 2000. New York, N.Y.: The Macmillan Company, 1967.

Laird, Donald A. and Eleanor C. Laird. How to Get Along with Automation. New York, N.Y.: McGraw-Hill Co., 1964.

Loughary, John W. Man-Machine Systems in Education. New York, N.Y.: Harper and Row, Inc., 1966.

National School Public Relations Association. Computers: New Era for Education? Washington, D.C.: The Association, a Department of the National Education Association, 1968.

Current developments in educational applications of the computer are discussed in this pamphlet. Included are the use and implications of CAI, computer instruction, computer counseling, paper marking and administration. Good updating for every teacher; fast reading.

National School Public Relations Association. Technology in Education. Washington, D.C.: The Association, a Department of the National Education Association, 1967.

A good pamphlet for updating teachers on the current status of educational technology. It is fast reading, with a good bibliography and is not just a listing of programs but a discussion of the technological implications in several areas.

Parkhill, Douglas F. The Challenge of the Computer Utility. Reading, Mass.: Addison-Wesley, Inc., 1966.

The computer utility discussed here is a general purpose public computer system in the same vein as power utilities and telephone utilities. The evolution, current status and future expectations of such computer utility systems are discussed along with a good chapter on time sharing technology.

Ross, Frank Jr. Automation, Servant to Man. New York, N.Y.: Lothrop, Lee and Shepard Co., Inc., 1965.

A general overview of the applications and implications of automation, written for young people. It is somewhat out of date, but gives an idea of the state of automation at the end of the last decade.

Seligman, Ben S. Most Notorious Victory. New York, N.Y.: Free Press, 1966.

The author surveys the history of modern technology including the development of computers, then discusses some applications and their social effects and the solutions proposed by government, labor and others. It is well researched and extensively documented, making a good source book. The author does not attempt to offer solutions to the problems of automation. Rather, he seeks to alert people to be concerned about the social consequences of automation and to stimulate readers to avoid the "Notorious Victory" of machines over men and assert human control.

Singh, Jagjit. Great Ideas in Information Theory, Language and Cybernetics. New York, N.Y.: Dover Publications, 1966.

Star, Jack. "The Computer Data Bank: Will It Kill Your Freedom?" Look, June 25, 1968.

Thomas, Shirley. Computers--Their History, Present Applications and Future. New York, N.Y.: Holt, Rinehart and Winston, Inc., 1965.

von Neumann, John. The Computer and the Brain. New Haven, Conn.: Yale University Press, 1958.

Westin, Alan F. Privacy and Freedom. New York, N.Y.: Atheneum, 1968.

A rational, scholarly, well-documented analysis of the problems involved in preserving individual privacy. One virtue of the book is that it is not an arm-waving exposition, but attempts to provide a rational foundation for the defense of personal privacy. The four parts of the book are The Functions of Privacy and Surveillance in Society, New Tools for Invading Society, American Society's Struggle for Controls and Five Case Studies and Policy Choices for the 1970's.

Wiener, Norbert. The Human Use of Human Beings. New York, N.Y.: Avon Book Co., 1967.

Ziegler, James R. Time-Sharing Data Processing Systems. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1967.

BUSINESS ORIENTED

American Institute of Certified Public Accountants. New Techniques for Business Planning, Systems and Control. New York, N.Y.: The Institute, 1967.

Awad, Elias M. Automatic Data Processing, Principles and Procedures. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1966.

Awad, Elias M. Business Data Processing. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1965.

Burck, Gilbert. The Computer Age and Its Potential for Management. New York, N.Y.: Harper and Row, Inc., 1965.

Cashman, Thomas J. and William J. Keys. Basic Projects in Data Processing. Anaheim, Calif.: Anaheim Publishing Co., 1967.

Cashman, Thomas J. Practical Projects in Data Processing. Anaheim, Calif.: Anaheim Publishing Company, 1967.

Cashman, Thomas J. Review Manual for Certificate in Data Processing. Anaheim, Calif.: Anaheim Publishing Company, 1965.

Chicago Board of Education. Curriculum Guide for Introduction to Data Processing, Grades 11-12. Chicago, Ill.: The Board, 1967.

Data Processing Management Association. Principles of Automatic Data Processing. Park Ridge, Ill.: The Association, 1965.

Elliot, C. Orville and Robert S. Wasley. Business Information Processing Systems. Homewood, Ill.: Irwin Company.

Freeman, Hanna and Hahn. Study Guides and Working Papers for Accounting 10/12, Parts I & II. New York, N.Y.: McGraw-Hill Company, 1968.

Gentle, Edgar C., Jr. Data Communications in Business: An Introduction. New York, N.Y.: American Telephone and Telegraph Co., 1965.

Gibson, E. Dana. An Introduction to Automated Data Processing. Elmhurst, Ill.: The Business Press, 1966.

Gregory, Robert H. and Richard L. Van Horn. Business Data Processing and Programming. Belmont, Calif.: Wadsworth Publishing Co., Inc., 1963.

Gregory, Robert H. and Richard L. Van Horn. Automatic Data Processing Systems, Principles and Procedures. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1956.

Haga, Enoch. Understanding Automation. Elmhurst, Ill.: The Business Press, 1965.

This is a book written specifically for teachers with the objectives of educating about general principles and giving assistance in the teaching of them. The first third of the book covers history, digital and analog principles, and the business of data processing. A useful historical time chart is in Chapter 1. The rest of the book is concerned with automation in secondary and higher schools with suggested course and curriculum outlines for high schools, junior colleges and colleges. Most helpful are the sections on an "Automation Day," business games, films and hardware. The Student's Manual for the above book contains, for each chapter, a list of the main ideas, a sample test and lists of questions and activities, all of which would be useful models for courses.

Kahn, Gilbert, editor. Business Data Processing: Basic Principles and Applications. New York, N.Y.: McGraw-Hill Book Co., 1966.

Langenbach, Robert G. Introduction to Automated Data Processing. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1968.

Martin, E. Wainright, Jr. Electronic Data Processing, An Introduction. Homewood, Ill.: Richard D. Irwin, Inc., 1965.

Rice, Harry G. and Jack W. Friedman, editors. Fundamentals of Electronic Data Processing. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1962.

Robichaud, Beryl. Understanding Modern Business Data Processing. New York, N.Y.: McGraw-Hill Book Co., 1966.

Roman, John C. The Business Curriculum. Burlingame, Calif.: Southwestern Publishing Co., 1966.

Saxon, James A. and Wesley W. Steyer. Basic Principles of Data Processing. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1967.

Schmidt, Richard N. and William E. Meyers. Electronic Business Data Processing. New York, N.Y.: Holt, Rinehart and Winston, Inc., 1963.

Solomon, Irving I. and Laurance O. Weingart. Management Uses of the Computer. New York, N.Y.: The New American Library, 1966.

Van Ness, Robert G. Principles of Data Processing with Computers. Elmhurst, Ill.: The Business Press, 1966.

Van Ness, Robert G. Principles of Punched Card Data Processing, Secondary Schools Edition. Elmhurst, Ill.: The Business Press, 1965.

Van Ness, Robert G. Principles of Punched Card Data Processing, Student's Manual. Elmhurst, Ill.: The Business Press, 1967.

Wanous, S. J. and Edward E. Wanous. Automation Office Practice. Cincinnati, Ohio: Southwestern Publishing Co., 1966.

Wenner, James F. A High School Orientation Course in Data Processing, Grades 11-12. Burlingame, Calif.: Southwestern Publishing Co., 1966.

MATHEMATICS ORIENTED

Arnold, B. H. Logic and Boolean Algebra. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1962.

Dorn, William S. and Herbert J. Greenberg. Mathematics and Computing: With FORTRAN Programming. New York, N.Y.: John Wiley & Sons, Inc., 1967.

A text for students with a background of two or three years of high school math. It is mainly a math text with computer programming viewed as a tool. It contains sections on linear equations and inequalities, iterative methods, infinite sums, Boolean algebra and probability. FORTRAN programming is introduced gradually, one full chapter is devoted to it and it is employed wherever the computer can be used to advantage. An excellent appendix discusses the relationships between monitors, compilers, assemblers and machine language. Annotated instructor's edition for the text includes the student's text with helpful comments, suggested quizzes, solutions to problems and quizzes and suggested references not in the student's text. Soon to be published is a new version of the book with BASIC programming.

Gruenberger, Fred and George Jaffray. Problems for Computer Solution. New York, N.Y.: John Wiley & Sons, Inc., 1965.

IBM Corporation. Computer Mathematics in Secondary Schools. Armonk, N.Y.: IBM Corp., 1966.

Jacobowitz, Henry. Computer Arithmetic. New York, N.Y.: John F. Rider, Inc., 1962.

This readable book is designed for instruction or review in basic arithmetical operations of number systems having applications in computers. Included are a survey of number systems, a detailed discussion of binary, octal, hexadecimal and ternary arithmetic and methods of conversion from one to another. Many examples are included and it could be a self-instruction manual. Two useful pages are given to illustrations of physical representation of binary data.

Johnson, David C. CAMP (Computer Assisted Mathematics Program). Glenville, Ill.: Scott, Foresman and Co., 1969.

> This is a set of texts for grades 7, 9 and 11. Chapter 1 is a good introduction to computers and flowcharting and is the same in all five books. Otherwise, each book in the set is different, dealing with the math topics at each level and employing the BASIC language. The texts are intended to teach math through computing and are a good source of ideas.

Johnson, Donovan A. and William H. Glenn. Understanding Numeration Systems. New York, N.Y.: McGraw-Hill Book Co., 1960.

National Council of Teachers of Mathematics. Careers in Mathematics. Washington, D.C.: The Council, 1962.

National Council of Teachers of Mathematics. Computer Oriented Mathematics, An Introduction for Teachers. Washington, D.C.: The Council, 1963.

National Council of Teachers of Mathematics. Formulas, Graphs and Patterns. Washington, D.C.: The Council, 1966.

Pennington, Ralph H. Introductory Computer Methods and Numerical Analysis. New York, N.Y.: Macmillan Co., 1965.

Price, Wilson T. and Merlin Miller. Elements of Data Processing Mathematics. New York, N.Y.: Holt, Rinehart and Winston, Inc., 1967.

> Topics suitable for mathematics courses in data processing curricula are the aim of this text. Prerequisite for studying the first half of the book is one year of high school algebra; and many topics are from second year algebra. Proof is minimized and problem solving emphasized. Numerical methods and Boolean algebra are included as well as an interesting section on sets using decks of punched cards to illustrate set concepts.

Quayle, James B. General Electric Computer Time Sharing in the High School Mathematics Curriculum. Cleveland, Ohio: Cleveland Information Processing Center, General Electric, 1967.

Ralston, Anthony and Herbert S. Wilf, editors. Mathematical Methods for Digital Computers. New York, N.Y.: John Wiley & Sons, Inc., 1967.

School Mathematics Study Group. Algorithms, Computation and Mathematics, Student Text. Palo Alto, Calif.: The Study Group, 1966.

School Mathematics Study Group. Algorithms, Computation and Mathematics, Student Text--FORTRAN Supplement. Palo Alto, Calif.: The Study Group, 1966.

School Mathematics Study Group. Algorithms, Computation and Mathematics, Student Text--ALGOL Supplement. Palo Alto, Calif.: The Study Group, 1966.

PROGRAMMING AND SYSTEMS ANALYSIS

ACM Professional Development Seminar. Information: Its Storage, Retrieval and Management. Philadelphia, Pa.: Auerbach Corp.

Anderson, Decima M. Basic Computer Programming. New York, N.Y.: Meredith Publishing Co., 1964.

Andree, Richard V. Computer Programming and Related Mathematics. New York, N.Y.: John Wiley & Sons, Inc., 1966.

Becker, Joseph and R. M. Hayes. Information Storage and Retrieval: Tools, Elements, Theories. New York, N.Y.: John Wiley & Sons, Inc., 1967.

Cutler, Donald I. Introduction to Computer Programming. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1964.

> An elementary text for students with a math background of a year of high school algebra, and with no access to a computer. After discussing number systems and a good chapter on flow charting, the author describes a hypothetical high-level language. The author assumes that the transfer to an actual machine and language will be relatively simple.

Farina, Mario V. Programming in BASIC. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1968.

Flores, Ivan. Computer Programming. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1966.

Flores, Ivan. Computer Software. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1965.

Hagg, James N. Comprehensive FORTRAN Programming. New York, N.Y.: Hayden Book Co., 1968.

Hull, T. E. Introduction to Computing. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1966.

In general the book deals with algorithms, their translation and use by computers and how computer logic affects problem solving. Chapters 5-10 contain a good description of FORTRAN IV as used with algorithms. The final third of the book is concerned with applications. It is not a book for beginners; and a reader should be familiar with college level math.

Hare, Van Court, Jr. Systems Analysis: A Diagnostic Approach. New York, N.Y.: Harcourt, Brace and World, Inc., 1967.

Kemeny, John G. and Thomas E. Kurtz. BASIC Programming. New York, N.Y.: John Wiley & Sons, Inc., 1967.

Laurie, Edward J. Computers and Computer Languages. Cincinnati, Ohio: Southwestern Publishing Co., 1966.

After the beginning general chapters in computers, punched cards and programming, this book takes up the IBM 1620 and 1401 computers. This is followed by discussions of SPS, FORTRAN, AUTOCODER, COBOL and the IBM 1130, finishing with some applications and comments on the future. It is intended as a text, but would also be a good classroom reference. The first three languages are compared by using the same sample problem to introduce them. Study guides for this book contain sets of objective questions and lists of important terms and concepts for each chapter.

Lecht, Charles Philip. The Programmers' ALGOL. New York, N.Y.: McGraw-Hill Book Co., 1967.

Luftig, Milton. Computer Programmer. New York, N.Y.: Arco Publishing Co., 1966.

McCracken, D. D. A Guide to FORTRAN IV Programming. New York, N.Y.: John Wiley & Sons, Inc., 1965.

Meadow, Charles T. The Analysis of Information Systems. New York, N.Y.: John Wiley & Sons, Inc., 1967.

Murphy, John S. Basics of Digital Computer Programming. New York, N.Y.: John F. Rider Co., 1967.

This book was written to provide the basic framework for learning computer programming. It attempts to distill that part of the programming background which is fundamental and common to all computers. It is recommended for preliminary study before approaching the study of a specific machine or programming language. The book is also intended for those who need a basic understanding of programming even though they may never need to practice the art.

Nathan, Robert and Elizabeth Hanes. Computer Programming Handbook: Guide for Beginners. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1961.

National Council of Teachers of Mathematics. Computer Facilities for Mathematics Instruction. Washington, D.C.: The Council, 1967.

Sharpe, William F. BASIC: An Introduction to Computer Programming Using the BASIC Language. New York, N.Y.: The Free Press, 1967.

The first half of the book describes BASIC and its use and contains a set of problems and solutions for each chapter. The last half has four appendices, two explaining the Dartmouth and University of Washington systems, one containing a short library of BASIC programs and one being a summary of the language. Some previous knowledge of flowcharting would be helpful to a reader since there is none included. The problems and library programs are generally oriented to economics and the social sciences rather than physical sciences.

Smith, Robert E. The Bases of FORTRAN. Minneapolis, Minn.: Control Data Institute, 1967.

Stark, Peter A. Digital Computer Programming. New York, N.Y.: Macmillan Company, 1967.

A text for the beginning programmer, including real problems presolved on a computer, and requiring little emphasis on mathematical abilities. Section 1 contains general information about computers and a chapter on flowcharting. Other sections are on machine, symbolic and problem oriented languages (especially FORTRAN) and programming and mathematical techniques. Worked examples are given complete with computer printout.

Wilson, Ira G. and Marthann E. Wilson. Information, Computers and System Design. New York, N.Y.: John Wiley & Sons, Inc., 1967.

SCIENCE AND TECHNICAL

Alcosser, Edward. How to Build a Working Digital Computer. New York, N.Y.: Hayden Book Co., 1967.

This is written for students with little or no computer background, but who are interested in the working of as well as programming of computers. It discusses each component of a computer, then gives construction details, resulting finally in a manually-operated digital computer that will demonstrate and perform fundamental computer operations. Included is a section on programming this machine.

Alt, F. and M. Rubinoff, editors. Advances in Computers. New York, N.Y.: Academic Press, 1967.

Bartee, Thomas C. Digital Computer Fundamentals. New York, N.Y.: McGraw-Hill Book Co., 1966.

Bartee, Thomas C. Digital Computer Fundamentals, Laboratory Manual. Cambridge, Mass.: Hickok Teaching Systems, Inc., 1964.

Benrey, Ronald. Understanding Digital Computers. New York, N.Y.: John F. Rider Publishers, Inc., 1964.

Some electronics background is helpful but not necessary for reading this book. It gives a general coverage of programming, number systems, logical design, circuitry and memory division. It is similar to the Siegel book by the same title in the information presented and in the profusion of useful illustrations and figures. Especially helpful are the machine language programming examples. Review questions follow each chapter and a glossary is included.

Bork, Alfred M. FORTRAN for Physics. Reading, Mass.: Addison-Wesley, 1967.

Burroughs Corporation. Digital Computer Principles. New York, N.Y.: McGraw-Hill Book Co., 1962.

Davis, Gordon B. An Introduction to Electronic Computers. New York, N.Y.: McGraw-Hill Book Co., 1965.

Digital Equipment Corporation. Small Computer Handbook 1968. Maynard, Mass.: The Digital Equipment Corp., 1968.

Harcourt, J. C. We Built our own Computers. Cambridge, Mass.: The University Press, 1966.

Written by high school age boys for that level, the book is good for those who wish to do more with computers than just program. Several examples of simple machines are given with directions for construction using inexpensive equipment, beginning with a matchbox computer and progressing to electrical ones. Sections include logic, binary arithmetic, analog computers and programming.

Ketchum, Donald J. Applications of Digital Logic. Farmingdale, N.J.: Buck Engineering Co., 1966.

Klerer, Melvin and Granino A. Korn. Digital Computer User's Handbook. New York, N.Y.: John F. Rider Publisher, 1967.

Murphy, John S. Basics of Digital Computers, Vol. 2 and 3. New York, N.Y.: McGraw-Hill Book Co., 1958.

Siegel, Paul. Understanding Digital Computers. New York, N.Y.: John F. Wiley & Sons, Inc., 1961.

This book, for readers with technical or scientific background or interests, presents the principles of computers in three sections. Section I includes logic and arithmetic with good tables and illustrations. Section II is concerned with the mechanical and electrical components. Section III discusses the functional units with a final chapter on a specimen computer. Each chapter has a concise summary and vocabulary list with a glossary at the end of the book.

Wiener, Norbert. Cybernetics, or Control and Communication in the Animal and Machine. Cambridge, Mass.: MIT Press, 1965.

MISCELLANEOUS

Finerman, Aaron and Lee Revens, editors. Comprehensive Bibliography of Computing Literature, 1967. New York, N.Y.: Association for Computing Machinery, 1968.

Holmes, James F. Data Transmission and Data Processing Dictionary. New York, N.Y.: John F. Rider Publisher, 1966.

Horn, Jack. Computer and Data Processing Dictionary and Guide. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1968.

FILMS

America on the Edge of Abundance. 60 min., 16 mm, sound, b & w. Audiovisual Section, Oregon State System of Higher Education, Portland, Oregon, 1965.

Summarizes some problems in American society brought on by automation, especially those of abundance and waste, and unemployment and the right to consume. The historical aspects of the work ethic are considered. Contains commentary by various experts on the above problems and on the future capabilities of computers.

Area of Action. 29 min., 16 mm, sound, color. IBM, Armonk, N.Y., 1960.

This film discusses EDP in industries of several countries: Australia--wood production; Taiwan--sugar processing; India--jute fiber quality control; Japan--Noritake dinnerware and Tokyo--banking; Thailand--statistical data for U.N. reports. Interesting, but does not show much software or hardware.

Automation. Reel 1--37 min., Reel 2--23 min., Reel 3--25 min., 16 mm, sound, b & w. Audiovisual Section, Division of Continuing Education, Oregon State University, Corvallis, Ore., 1957.

Narrated by Edward R. Murrow, the film explores the effects of automation in many industries. Although a decade old, many points are pertinent today. Takes more than one class period, but continuity is not harmed by a break between reels. Reel 1: Comments by Walter Reuther and John Diebold. Discusses development of automation from James Watt on. Examples shown: aircraft production, auto engines, steel, air traffic control, cancer detection. Reel 2: Comments at a union meeting on the effect of automation on workers, and their solutions. Comments by T. J. Watson, President of IBM. Examples shown: telephones, light bulb production, bakery, home freezer-stove combination. Reel 3: Comments by Walter Reuther and an MIT professor. Interesting points: We need a clarification of values. We should share abundance rather than divide scarcity. Automation is an intellectual revolution. Increasing the speed of communication doesn't improve that which is communicated.

Automation: The Next Revolution. 30 min., 16 mm, sound, b & w. Audiovisual Section, Division of Continuing Education, Oregon State University, Corvallis, Ore., 1966.

Good discussion starter. Includes comments by business and labor leaders on their views of the problems of automation, and some examples of automation such as shipping company and bottling plant. Comments by Willard Wirtz and Bayard Rustin are interesting and provocative. Some questions considered: Is there always a place in society for a man who can earn his way? Do events force businessmen to automation? Should the rate of introduction of machines be such as to protect existing workers? Should we separate pay from work? The narrator's final comment is that the problems will be solved by inventiveness, courage and kindness, which machines do not possess.

Class of 01: The College of Tomorrow. 25 min., 16 mm, sound, color. McGraw-Hill Book Co., New York, N.Y., 1967.

Narrated by Walter Cronkite, this film summarizes the new educational technology, with commentary on the future outlook from Harold Howe and Fred Heckinger. A couple of interesting points: "Technology will become not only an adjunct but a necessity," and "Teachers will 'teach' less so that learners can learn more." Some items shown are the multi-media center at Pennsylvania State, the use of a learning package with tapes, slides, films, etc., for individual study, and a demonstration of a lab experiment conducted by computer and displayed on a video screen (an application of PLATO--Programmed Logic for Automatic Teaching Operation).

Communications Explosion. 29 min., 16 mm, sound, color. Marion County, Oregon, Intermediate Education District, Salem, Ore., 1967.

Narrated by Walter Cronkite. Not entirely devoted to computers, the film does show their application and importance to communication. Orbiting satellites are discussed by Arthur C. Clark, one of the originators of the synchronous satellite concept. An interesting point was that improved communications could reduce the need for business travel. Other interviewers are John R. Pierce on laser transmission and John Diebold on self-edited news reports and man-machine interface. Some topics included: conversations with computers through terminals, oral conversation with a computer, computer generated voice.

Computer Programming. 33 min., 16 mm, sound, b & w. System Development Corp., Santa Monica, Calif., 1958.

Although it is an old film, it is still worthwhile and it is preceeded by a leader which updates it somewhat. The programmer's task is described. A brief discussion of real-time is followed by a sample problem--the avoidance of objects in space by a space vehicle. A definition of a program is given, as well as a discussion of procedures prior to writing a program, including flowcharts with decisions and loops. Also shown are the relationship between flowchart and program and the processes of testing and debugging. All of these are with reference to the sample problem. The programmer is shown as a member of a team, coordinating his part with the larger problem.

The Computer Revolution. 22 min., 16 mm, sound, color. CBS Television Network, New York, N.Y., 1966.

Narrated by Walter Cronkite, taken from "21st Century" on CBS-TV. This is a fast-paced film introducing the viewer to computer applications in the following areas: newspapers, air traffic control, hospital information, space flight control, police work, graphic display, voice simulation, chess, national data bank and time sharing. Some comment is made on possible safeguards for privacy. A good introductory film, it does not go into detail on any one application.

Computer Technique for Animated Movies. 17 min., 16 mm, no sound, b & w. Pacific Northwest Bell Co., Portland, Ore.

A slowly paced film with no sound. It describes the process of producing an animated film. The desired animation is programmed using a language developed for the purpose, and the program is fed to a computer which produces a control tape. The tape then is used to control film motion in the camera and produces the visual display on a video tube, at which the camera is aimed. This film is made by the technique it illustrates. An added section shows an optical paradox filmed this way.