Richard Mowe and Ron Mummaw

ACADEMIC
T I®

THE ACADEMIC TI

THE ACADEMIC TI

Richard Mowe
Ronald Mummaw

RESTON PUBLISHING COMPANY, INC.
A Prentice-Hall Company
RESTON, VIRGINIA

Library of Congress Cataloging in Publication Data

Mowe, Richard.
The academic TI.

"A Reston Computer Group book."
Includes index.
1. Computer-assisted instruction. 2. TI 99/4A
(Computer)—Programming. I. Mummaw, Ronald. II. Title.
III. Title: Academic T.I.
LB1028.5.M68 1984 371.3'9445 84-4736
ISBN 0-8359-0021-5
ISBN 0-8359-0020-7

A Prentice-Hall Company
Reston, Virginia 22090

10 9 8 7 6 5 4 3 2 1

PRINTED IN THE UNITED STATES OF AMERICA

To Teri Perl

CONTENTS

Acknowledgments xi

Preface xiii

Chapter 1

INTRODUCTION 1

Who Should Read This Book? 2
Knowledge Required 2
Equipment and Software Needed 3
How to Use This Book 5
A Note to Parents 5
A Note to Teachers 6
A Note to Students 7

Chapter 2

GETTING STARTED 9

Parts of a Computer 9
Hardware 9

Software 13
Steps in Using the Computer 13
What to Expect at Different Age Levels 16
Things to Encourage 23
Things to Discourage 23

Chapter 3
USING COMMERCIAL SOFTWARE 25
Types of Software 27
Characteristics of Good Educational Software 29
Software Frustrations 32
Software Advice 33

Chapter 4
DOING PROGRAMMING 35
Importance of Programming 35
Programming Languages 36
Choice of a Programming Language 38
Learning to Program 40
Displaying Information 42

Chapter 5
PROGRAMMING IN BASIC 44
Lesson Format 44

Chapter 6
PROGRAMMING IN LOGO 78

Chapter 7
WRITING SOFTWARE 91
Getting Started 93
Designing Phase 95
Programming Phase 98
Testing Phase 101
Using and Modifying the Programs in This Book 102
Sample Programs 103

Chapter 8

LEARNING TO TYPE 130

Why Learn to Type? 131
When to Learn Typing 132
How to Learn 133
Using Touch Typing Tutor 134

Chapter 9

USING A WORD PROCESSOR 136

What is Word Processing? 136
Who Uses Word Processors? 137
How Are Word Processors Used? 138
Advantages of Word Processing 139
Disadvantages of Word Processing 140
Learning Word Processing 141
Using the TI as a Word Processor 142
Using the Lessons 143
Helpful Hints 158

Chapter 10

SUMMARY 160

Getting Started 160
Using Commercial Software 161
Doing Programming 162
Writing Software 162
Learning to Type 163
Using a Word Processor 163
Closing Thoughts 163

APPENDICES 164

Helpful Forms 164
Helpful Books 167
Helpful Magazines 169
Sources of Software Evaluations and Evaluation Forms 170
Software Recommendations 171
Glossary 173

Index 174

ACKNOWLEDGMENTS

We could not have written this book, if it had not been for the help of others. Teachers and administrators of the Muroc Joint Unified School District have helped and encouraged both of us.

Ron and I are further thankful for the opportunity to refine our programming worksheets by teaching in the STEP program at Antelope Valley College.

Les Merryman was helpful in answering technical questions about the TI99/4A®.

We would also like to thank Dwight Stapley, the principal of Joshua Elementary School, for lining up students to pose for pictures.

Finally, I thank my students at Desert High School for inspiring ideas and posing for pictures, and you, Danny Holm, for developing pictures on short notice.

PREFACE

We have written *THE ACADEMIC TI* for people who want to learn with the aid of a computer. Teachers, parents, and others who work with young people will find the book most useful.

This book is divided into three general sections. The first section includes introductory material about computers and the use of computers with young people. The next section covers the use of software. The following section teaches programming. The last section discusses learning to type and using a word processor.

We tried to keep the vocubulary as simple as possible for computer beginners. For those of you with computer experience, we have included material that will help you introduce computers to young people.

Problems and exercises will be included that require a TI 99/4A computer. Some activities will require a disk drive and printer. Most will not. If you do not have access to a computer, reading this book can still help you learn about computers in education.

Richard Mowe
Ronald Mummaw

chapter 1 INTRODUCTION

A Texas Instruments (TI) 99/4A computer can help you learn. As a learning aid, the computer is patient beyond belief. It won't get tired, have to take a break, or get irritated when you make a mistake. It can help you practice basic skills using appropriate software. The computer will present work at your own level and keep an accurate record of your progress. Not only that, but you probably also will enjoy doing your work.

When you get tired of practicing skills on the computer and want to write your own programs, your TI is there to assist you. This book will give you programming assignments, but the way to learn to program is to try programming yourself.

A computer can help you work more efficiently. One area of increased efficiency is writing. With a word processing program, you can type text into the computer. Then you can edit right on the screen. You can easily add and delete words and correct typographical errors. When everything is satisfactory, a computer printer will print the text.

WHO SHOULD READ THIS BOOK

If you're in one of the following categories, you should read this book.

- Parents. You own a TI computer or are considering buying one and would like to help your children receive a better education. Perhaps a son or daughter needs help with math. Another is bored and needs a challenge.
- Teachers. Your school just bought a new computer. You have heard that schools are starting to use computers and that many parents are buying personal computers. You're interested in learning how computers can help your students learn, both in the classroom and at home.
- Students. You have access to a TI computer and would like to learn to use it. You're good at learning on your own or know someone who can help you.
- Anyone. You have an interest in exploring the field of computers in education.

KNOWLEDGE REQUIRED

This book is for beginners. You're either a computer beginner, a beginner at working with children and computers, or both.

For computer beginners, I have kept this book as simple as possible and included a glossary of computer words. As you work through the programming exercises in this book you will gain confidence and feel more comfortable with the computer. Typing programs into your TI computer will give you added practice working with a computer.

Perhaps you're already familiar with computers and can program well. You will learn how to introduce computers to young people. Different aged students will be ready to learn different things. They'll need to use different software and will be ready to learn different programming skills.

You must realize that it is okay to make mistakes. Working with computers is frustrating at times, and things won't always go the way you expect. Just keep on trying and keep your sense of humor.

EQUIPMENT AND SOFTWARE NEEDED

To take maximum advantage of this book, you'll need to have access to a computer system. However, you can learn a lot just from reading and can get ideas on what computer to buy. You also can read this book to see what can be done with a computer in education.

This book includes discussion of the components you'll need to complete the activities in this book and the requirements for the various activities. You won't need the same equipment for all activities. Generally, you'll need the most equipment for word processing and the least to learn elementary programming. The following paragraphs discuss equipment you'll need.

■ Equipment

Console. The TI 99/4A is recommended for all the activities in this book. The TI 99/4 isn't compatible with TI-WRITER, the word processing program used in Chapter 7. The 99/4 will work for BASIC and Logo programming, but you'll have to make some modifications.

Video Display. To see the information the computer produces, you'll need a video display screen. Most people use their home television. A color television will take advantage of the color capabilities of the TI. A black and white television, however, will work fine.

As they do more programming and word processing, some people switch to a color monitor. An advantage of a monitor is that it is easier to read, because the characters show up more sharply. The TI color monitor comes with a special cord needed to connect a monitor to the TI console. If you buy another monitor, you'll have to buy the special cord.

If you use your TI primarily for word processing, then a monochromatic monitor, such as a green screen monitor, produces the sharpest screen image, although it lacks both color and sound features.

Peripheral Expansion System. If you're planning to add extra memory, a disk drive, or a printer, you'll need the peripheral expansion system. Its function is to hold and interconnect the various peripherals.

Mass Storage. When loading or saving programs, you'll need a mass storage device. Two common storage devices are the cassette tape recorder and the disk drive. The cassette tape recorder is the least expensive option. The TI program recorder works well and is supplied with the necessary connecting cord. The program recorder also doubles as a conventional tape recorder.

If you use a regular cassette recorder, make sure it's compatible with the console. You'll find a partial list of compatible recorders in the audio cassette recorder information sheet packed with the console. You'll need to buy a connecting cable.

For certain applications, such as word processing, the disk drive is necessary. Since the disk drive loads and saves programs almost 10 times faster than the cassette recorder, the disk drive is useful when you need to load and save programs quickly.

The disadvantage of the disk drive is its high cost. Besides the disk memory drive, you'll need the peripheral expansion system, disk drive controller, and 32K memory expansion.

Printer. To use a word processor, you need a printer. A printer is useful for other applications as well. If you're using a program that keeps records, a printer gives you a copy of the records. Another use is as an aid to programming. When you're writing a program, you can get a hard copy of the program to edit before you reenter the program into the computer.

In the lower price line, there are two types of printers to consider. The thermal printer is the least expensive and is most compact and quiet. Thermal paper, however, is expensive and sometimes hard to read.

The most popular printer is the dot-matrix type. It is inexpensive, uses ordinary printer paper, and is fast. The disadvantage is that it produces characters that are not of typewriter quality.

The most expensive type of printer is the letter-quality printer. It is sometimes called a daisy wheel printer. It produces typewriter-like characters, but has disadvantages: It's slow and lacks graphics capabilities.

Whatever printer you choose, you'll need the RS 232 interface and the peripheral expansion system.

Wired Remote Controllers. These are also called joysticks. Some programs use them.

Software

To do the word processing lessons in this book, you'll need the word processing program TI-WRITER. To do Logo programming, you'll need either TI-LOGO or TI-LOGO II. Other software will be discussed in this book, but isn't necessary.

HOW TO USE THIS BOOK

The purpose of this book is to help people learn. Learning skills in several areas are presented.

Chapter 2 contains an introductory section that will help you set up the computer. If you already know how to operate a computer, you'll want to skip part of the first section.

Chapter 3 discusses commercial software. There's a bewildering array of software. This section will help you sort out the different types of software and give suggestions for software selection.

Chapters 4, 5, and 6 teach programming. Learning programming will help you learn problemsolving skills and become better organized in your thinking.

Once you learn to program, you can learn to write your own educational software. In Chapter 7, you'll go through the design, programming, and testing stages of software design.

Chapter 8 gives hints on learning to type using a computer.

Finally, Chapter 9 shows you how to use a word processor and discusses ways to improve writing skills.

A NOTE TO PARENTS

Teaching your son or daughter to use a computer can be an enjoyable experience. You'll be teaching a valuable work skill and spending time together.

Each person is a unique individual and has unique educational needs. One person will work well with software already written. Another may want to write his own programs. Yet another may not want to use the computer at all.

Be sensitive to the needs of your son or daughter and use the sections that will be of benefit. Be patient. A computer can intimidate someone who has never used one.

If you're a computer beginner yourself, don't worry. You and your child can learn together.

A NOTE TO TEACHERS

You're a valuable person. I am glad you don't feel threatened about a computer taking over your job and are willing to learn about computers.

Before you use a computer with your class, you'll want to familiarize yourself with computer operation. A good place to start is with commercial software. All you'll have to do is load a program into the computer and follow the directions on the computer screen. After you feel comfortable with the computer, try the chapter on programming or word processing.

There are a lot of ways to use computers in the classroom. Make sure that you don't try to do too many things at once. Work on one area at a time. Writing a few objectives will help you focus on what is important.

After you've familiarized yourself with the computer and written your objectives, you'll need to gather teaching materials. I'll give you sources of commercial software and provide programs you can type into the computer. You have permission to duplicate the programming worksheets.

As a teacher, you'll have to consider group aspects of working with the computer. If you have only one or two computers in your classroom, you'll need to carefully consider who uses the computer and when it's used.

Where will you put the computer? Obviously, it should be close to an electric outlet. Some classrooms have only one outlet in the room. If possible, place the computer where you can see it, but where the students cannot, so that they won't be distracted.

When will the computer be used? If you have three or more computers, perhaps all the students in the class can use the computer in one class period as long as the assignment is short. With fewer computers, other scheduling may be better.

In a self-contained class, you can explain an assignment and schedule computer time throughout the day. In a departmentalized class, you can explain the assignment and schedule time throughout the week. If you have one computer and want all your students to use the computer in one period, give a short assignment or have the students work at the computer in groups of two to four.

In what order will students use the computer? One good way to schedule students is to go by rows. Another way is to make up a list.

Some teachers like to use the computer as an incentive for students to finish work in other subjects. If you let students use the computer only after they finish a regular assignment, make some provision for slower students to use the computer, too.

What should students do when they need help using the computer? One solution is to assign students to the computer in groups of two. A lot of computer mistakes are simple errors, such as forgetting to press the ENTER key. Often a partner will catch mistakes. Another way to provide help is to use a student helper. In every class, you'll find one or more students who catch on quickly or have a computer at home. They're usually more than happy to show off their knowledge and help classmates.

At first you may feel uncomfortable using the computer, but keep at it. Your class will be excited to use the computer and will profit from the experience. Since they'll be working at the computer by themselves or in small groups, they'll develop independent thinking. As they master computer skills, they'll improve in confidence and possibly carry positive attitudes gained from working with the computer into other classes. You already know that young people don't like drill. With the computer, drill can be fun.

A NOTE TO STUDENTS

If you're using this book on your own, here are two suggestions. First, learn as much as you can about programming the computer. It's fun to use someone else's software, but writing your own software will provide lasting benefits.

First, you'll learn a job-related skill. More and more occupations make use of computers. Learning to program now could give you a real advantage later when you enter a career.

Second, a knowledge of programming can help you in college. Computer study is required for some majors in college. If you learn to program now, you'll have an easier time later on.

Third, you'll develop a worthwhile skill and gain the confidence to tackle other areas that give you trouble.

Try to find someone to help you when you run into trouble. Teachers, other adults, and other students who have computer knowledge can help you. If possible, join a computer club. Clubs sometimes offer classes at little or no cost, and of course, club members like to talk about computers. To find a club, check with your computer dealer.

chapter 2 GETTING STARTED

PARTS OF A COMPUTER

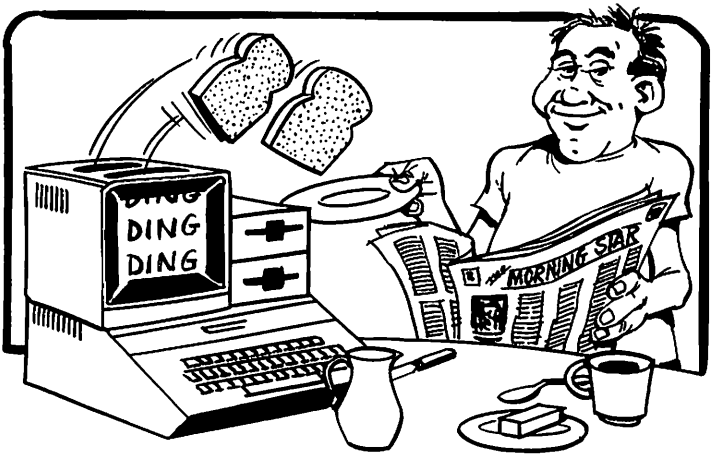
A computer system can be divided into two main categories. The first is hardware, or the physical parts of the computer system. Hardware includes the computer, screen, and disk drive. The programs that control the computer are called software. Programs may be recorded on modules, cassettes, and disks, or entered from the keyboard.

HARDWARE

Many people think of a computer as a mystical machine endowed with magical powers or a blob that's taking over their jobs. Neither conception is correct. A computer is merely a machine, and a dumb one at that.

One way to better understand a computer is to compare it to everyday objects. Think of a computer as a toaster. When we use a toaster, we put bread into the toaster. The bread is the input to the

FIGURE 2-1



Think of a computer as a toaster.

toaster. Next, we turn on the toaster. This is called processing. Finally, the toast pops up (if you remembered to plug in the toaster). The toast is the output.

Like the toaster, the computer has input, output, and processing. Input devices on a computer allow information to be fed into the computer. One input device is the keyboard. As you type on the computer, information enters the computer in coded form.

In many respects, the computer keyboard is like a typewriter keyboard. Most of the keys are in the same place. A computer has a few extra keys, which we'll discuss later.

An input device used in many computer games is the wired remote controller, also called a joystick. As the joystick is moved, an electric signal is sent to the computer. An example of a game using the joystick is MUNCHMAN.

Information leaving the computer is the output. Most of the time, output is displayed on the screen. The screen is either a television or a video display monitor. The screen also is called a cathode ray tube or CRT, the picture tube in a television or monitor.

Some components perform both input and output functions. They are called I/O devices. One example is the disk drive. If you're using a disk drive to load a program, for instance a word processor, then you're using an input device. The disk drive is used for output when you save a word processing text file.

The cassette tape recorder is an inexpensive I/O device. The recorder loads and saves programs also, but at a much more leisurely pace than the disk drive.

The main processing of the computer takes place within the computer itself. The secret of the microcomputer's power and compactness lies in the rectangular components called integrated circuits, or "chips." The most important chip in the computer is the microprocessor.

The microprocessor performs several tasks vital to the functioning of the computer. First, it decodes and carries out instructions. You understand instructions in English, such as "take out the garbage." The microprocessor understands instructions written in machine language. Machine language consists of instructions written in binary, or base 2, code.

Binary numbers consist of 1s and 0s. For instance, the number 13 in base 2 is 1101.

Next the microprocessor controls other parts of the computer system. For instance, if you insert a program module while the computer is turned on, the processor sends out a signal that resets the computer and displays the master title screen.

Another function of the processor is to handle arithmetic and logical tasks. The processor can handle many additions and subtractions per second. Logical operations include comparisons and decisions.

Another major part of the computer is its memory. You remember instructions and so does the computer. The 99/4A contains two types of memory. The first type of memory is the read only memory (ROM).

Read only memory contains the language of the computer that's easier for us to understand than 1s and 0s. A computer language we'll be studying in another chapter is called BASIC, which stands for Beginners All Purpose Symbolic Instruction Code.

The computer can read information in the ROM but cannot change the ROM. That's why it's called a read only memory.

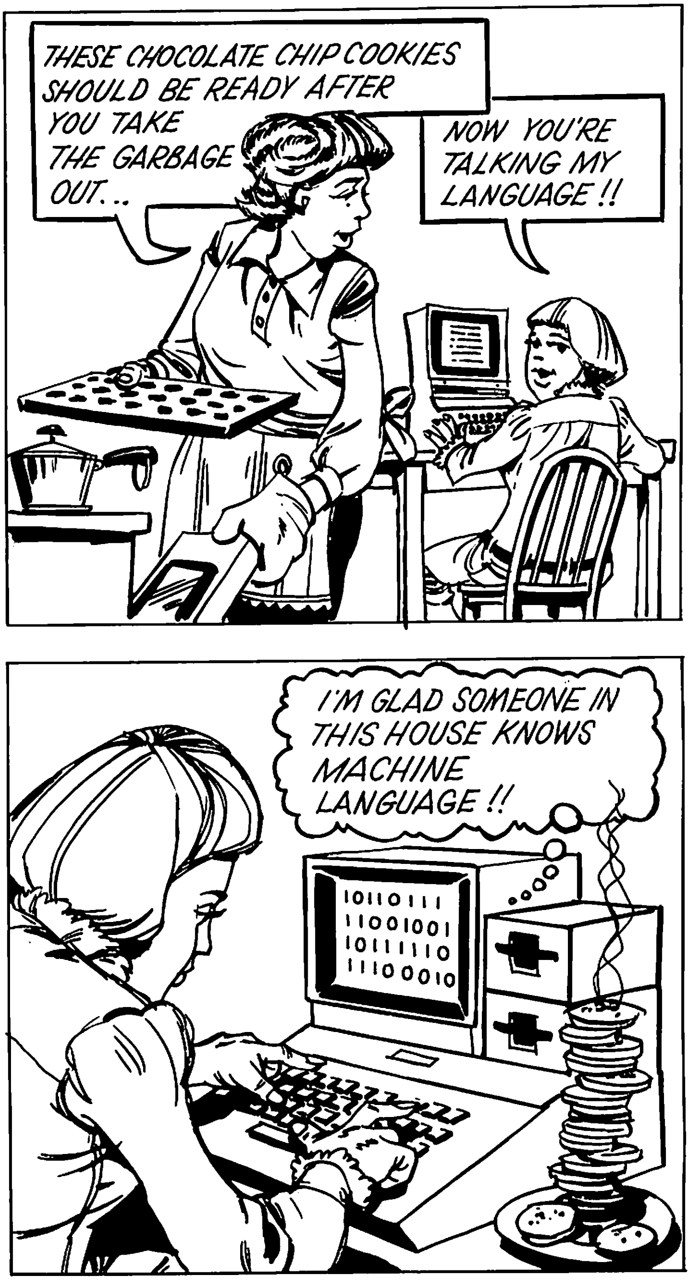
The other type of memory is the random access memory (RAM). The computer can both read information from the RAM and write new information in it. The RAM has one drawback, however. It can store information only as long as the computer has electricity. When the computer is turned off, the RAM loses the information stored in it.

Computers vary in the amount of memory they contain. Memory is measured in units called bytes. Each letter, number, character, and space is stored as a byte of information in the computer's memory.

A more useful unit of memory is the kilobyte, or K for short. 1 K = 1,024 bytes. The TI 99/4A console contains 16K of RAM. The peripheral expansion system can hold another 32K of memory.

Accessories that plug into the 99/4A are called peripherals. Some peripherals, such as the speech synthesizer, plug into the console. Others, such as the disk drive, plug into the peripheral expansion system.

FIGURE 2-2


THESE CHOCOLATE CHIP COOKIES SHOULD BE READY AFTER YOU TAKE THE GARBAGE OUT...
NOW YOU'RE TALKING MY LANGUAGE !!
I'M GLAD SOMEONE IN THIS HOUSE KNOWS MACHINE LANGUAGE !!
10110111
11001001
10111110
11100010

Instructions human understands compared to instruction 6502 microprocessor understands.

SOFTWARE

Software contains instructions that control the computer and make it appear to be intelligent. Without software, a computer is useless. The instructions are organized into units called programs.

We compared a toaster to a computer. A toaster, like a computer, needs outside help to perform its job. Here is a program for making toast:

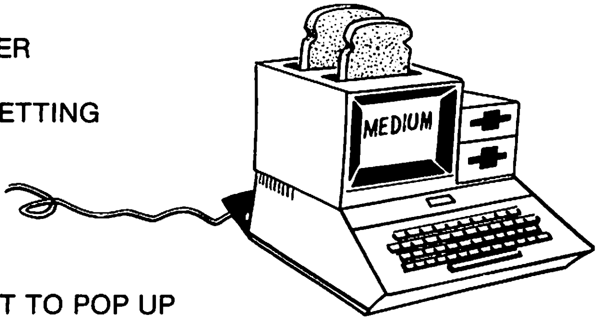
PLUG IN TOASTER

ADJUST HEAT SETTING

INSERT BREAD

PUSH LEVER

WAIT FOR TOAST TO POP UP



This book discusses three types of software. The first type is software that someone else writes to help you learn a skill. Some common types are tutorials, drills, simulations, and games.

Another type of software is the type you write yourself. Writing software involves programming the computer.

A third kind of software is meant to make you more productive or creative. A prime example is word processing software.

STEPS IN USING THE COMPUTER

The first step in using the computer is to connect the components properly. If you're a computer beginner, you'll probably want your computer dealer or some other knowledgeable person to show you how to set up your TI. If you have no one to help you, follow the directions in the manuals that came with your system.

After the computer is set up, you're ready to use it. The two main modes for using the computer are running software and doing programming.

Running Software

The easiest way to use your computer is to run software. The main ways to load a program are from module, tape, and disk.

Loading a Module. First, touch a metal object to discharge any electricity that might damage the module. Then turn on the peripherals (if you have any) and the television or monitor. Next turn on the console and wait for the master title screen to show up. Finally, insert the program module into the console.

Press any key, and the computer will display the master selection list. Press the number key that corresponds to the program you wish to use.

Loading a Cassette. If you're using the TI program recorder, set the volume and tone so that the white band is in the middle. If you're using another cassette tape recorder, read the audio cassette recorder information sheet packed with the console. Turn on the peripherals, television or monitor, and the console. Make sure the ALPHA LOCK key is down. Press any key, and when the next screen appears, choose 1 for TI BASIC. Now type OLD CS1 and press ENTER. Don't forget the space between OLD and CS1. From here on out, the computer will prompt you with the following statements.

```
REWIND CASSETTE TAPE
THEN PRESS ENTER

PRESS CASSETTE PLAY
THEN PRESS ENTER

READING
DATA OK

PRESS CASSETTE STOP
THEN PRESS ENTER
```

Be sure to press ENTER each time the computer requests it, because the computer controls the tape recorder motor.

The process of loading a tape can take several minutes. When the blinking cursor appears at the bottom of the screen, type RUN and press ENTER.

If the program doesn't load, the computer will display one of two error messages. If the error displayed is ERROR-NO DATA FOUND, try increasing the volume slightly on the tape recorder. Then press R to reread the tape.

If the computer gives the message ERROR IN DATA DETECTED, lower the volume slightly and press R.

Don't be surprised if you have to try to load several times. Volume and tone settings on the TI are critical. If you continue to have problems, consult the manual for the TI program recorder.

Loading a Disk. If your program doesn't use a module, start with the ALPHA LOCK key down. Otherwise, follow the module directions. Remove the disk from its protective sleeve. Be careful not to touch the exposed part of the disk. Hold the disk so that you'll insert it, with the label facing the side of the drive opposite the red light. (Hold the disk with your index finger on the label, if you're using a disk drive in the peripheral expansion system. If you are using an external drive, hold the disk with your thumb on the label.)

Next, open the disk drive door. Insert the disk gently, so that it doesn't bend, and push it until it stops. Now close the disk drive door.

Turn on the peripherals, television or monitor, and the console, in that order. If you're using TI-WRITER or another program that uses a module, insert the module now. Press any key to display the selection list. If you're using a program with a module, choose the program listed on the selection list. If not using a module, choose TI BASIC. When in BASIC, you should see the flashing cursor.

If your program doesn't use a module, the procedure is slightly different. Let's say the name of the program you want is XDRILL1. To run the program, type LOAD DKS1.XDRILL1 and press ENTER. I'm assuming you have one disk drive, or you have the disk in drive one.

If you've forgotten the name of the program, then you'll have to use the DISK MANAGER module to display the disk catalog. For more information on using the disk, consult the TI disk memory system manual.

■ Programming

To write your own programs, start with the ALPHA LOCK key down. Turn on the peripherals, television, and console. Press any key and then choose TI BASIC. When the blinking cursor appears, you can start programming.

WHAT TO EXPECT AT DIFFERENT AGE LEVELS

Computers are confronting us so rapidly that it's difficult to assess their role in education. A few years ago, most of us didn't see computers in the schools.

It's becoming more and more obvious that our society lacks computer knowledge. If you have any doubts, think of the difficulty in getting a computer billing error corrected.

As citizens in a computerized society, we need to know something about a computer and be able to use a computer. Knowing something about a computer and being able to use one are called computer literacy. As you read this book, you'll become computer literate. First you'll learn about the computer. You will know something about computer parts and see some applications in which a computer is used. Second, if you do the computer exercises, you'll increase your skill at using the computer.

The question is "what level of computer literacy is appropriate for different age groups?" For instance, at what age should children be introduced to the computer? What is a good age for starting programming? At what age should students learn word processing?

Curriculum guides have been developed for subjects such as reading and math. Few computer guides exist, however, because computers haven't been used widely in schools. There's a lot that we don't know about computers and young people. We do know, however, that young people are interested in computers. Just watch them playing video games at the local arcade. As interest grows in computers, educational uses for computers will expand.

Two things can help in planning computer activities. The first is a knowledge of student capability. Understanding students of a given age will help in designing software and activities for their needs. The other thing that will help in planning computer activities is to see some sample activities that can be used at each grade level. A word of caution is in order, however. Just as students vary greatly in reading ability, they'll vary in computer ability. Some third graders are able to program the computer, while some high school students never develop the ability.

■ Expectations for Kindergarten

Abilities.

- Reading. Readiness activities such as learning the letters (both upper- and lowercase) are taught. Other skills include recognizing letter sounds and shapes related to letters. By the end of the year, some are reading.

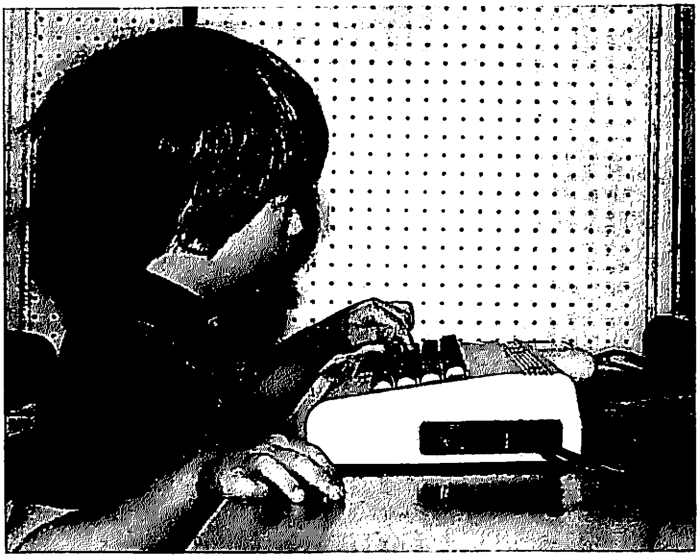
- Math. Numbers up to 30 are learned in oral and written form. Students should recognize the circle, square, triangle, and rectangle. Positions such as left, right, and middle are important. These students study attributes such as short, tall, fat, and thin.
- Language. Students should be able to print their name and follow simple directions. Activities such as "show and tell" give students an opportunity to express themselves.

Attention Span. After five to 10 minutes, the kindergartner is ready for a new activity. These activities include physical, mental, and social experiences.

Interests and Activities. Improving large motor skills is important. Activities include running and jumping. Students learn to skip, gallop, and jump on one foot. Dressing skills involve tying shoes, zipping, and buttoning.

Computer Activities. Programs on letter and number recognition work well. Other possibilities include shape recognition, as long as the programs do not involve reading. By the end of the year, some students can add simple sums, such as 2 + 3.

FIGURE 2-3



Kindergarten student at a computer.

Expectations for Grades One to Three

Abilities.

- Reading. Students learn and develop reading skills. Many teachers teach phonics. More and more "sight words" are added to the vocabulary. Students look for the main idea and the sequence of the story. Prefixes, suffixes, compound words, and root words are important.
- Math. Addition and subtraction skills at increasing difficulty levels are learned. Students work with the value of coins and making change. Measurement skills in both English and metric are taught. Students learn to recognize and write the values of simple fractions. Telling time is important. By the third grade, students perform simple multiplication and division problems.
- Language. Students learn and develop writing skills. They work on capitalization, punctuation, and abbreviations. They study nouns and verbs. Word usage and spelling are important. Students are starting to put sentences together to form paragraphs. They write letters.

Attention Span. Students are able to work a longer time on one task. There's still a need to vary activities. Some students have difficulty settling down to work.

Interests and Activities. Students develop varied interests. Some join team sports, others join organizations such as scouts. As they grow older, interests are directed outside the home.

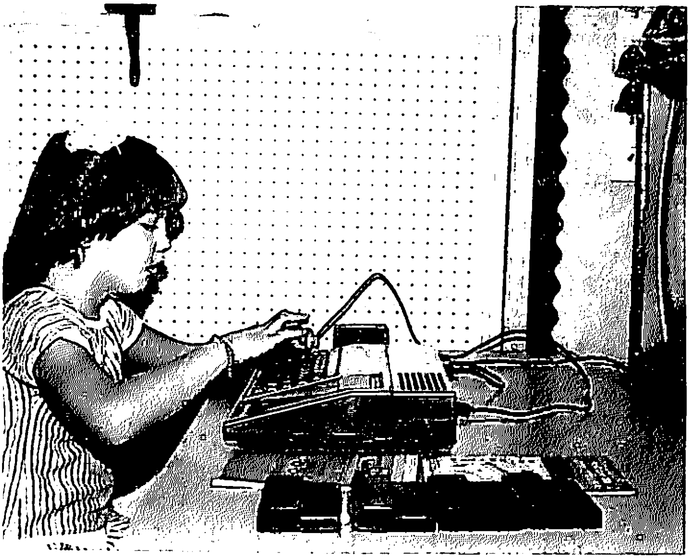
Computer Activities. This is a good age to start using drill and practice software. Students are learning a number of skills that need repetition. Possible drill areas are basic math facts, punctuation, prefixes, and suffixes. Students enjoy drill in a game format. Thinking games that are appropriate to this age level are good. Some students will enjoy LOGO programming.

Expectations for Grades Four to Six

Abilities.

- Reading. By the fourth grade, emphasis shifts from reading words to understanding what is read. Finding the main idea and sequence of the story is important. Thinking skills are further developed, for

FIGURE 2-4



A second grader.

example, making inferences, drawing conclusions, and determining cause and effect. A wider range of literature and poetry is studied.

- Math. Proficiency is gained in multiplication and division. Operations of fractions are studied—first addition and subtraction, later multiplication and division. Geometry concepts include angle measurement, polygons, and planes. Students work with decimals in more detail and are exposed to percents. Measurement concepts are extended and word problems are more complex.
- Language. Capitalization, punctuation, and abbreviations are still important. Nouns, verbs, adjectives, adverbs, objects, and pronouns are studied in detail. Research, library, and dictionary skills are taught. Students write longer reports and creative writing assignments.

Attention Span. This skill continues to improve from the primary grades, but individual differences continue to be great. Some do well scheduling their own time. Others need help.

Interests and Activities. Team sports become even more important for some. Activities outside the home continue to increase. Social awareness and peer pressure increase.

FIGURE 2-5

Fifth grader

Computer Activities. A large amount of software is available for this age group. Interesting drill programs are appropriate both for developing skills and providing remediation. Thinking games work well. Students are able to write short programs in BASIC and LOGO. Word processing is a possibility.

Expectations for Junior High

Abilities.

- Reading. Developmental reading, with the emphasis on comprehension, is taught to students at a low reading level. Finding the meaning of words in context is also stressed. Synonyms, antonyms, and homonyms are discussed.
- Math. Some students are still trying to master basic math skills. Others are studying algebra. Formulas, equations, and variables are studied. Geometry is given greater emphasis. Scientific notation is introduced. Word problems that involve more than one operation are given.
- Language. Effective communication and expression in speaking and writing are stressed. All parts of speech are covered. Students

practice diagramming sentences. There is greater emphasis on essays and creative writing. Students develop speaking skills.

Attention Span. In general, there is a greater ability to work independently on a given task.

Interests and Activities. Junior high students are at an age between childhood and adulthood. They are developing independence. Sports continue to be important. School activities such as dances are enjoyed. Peer relationships become extremely important.

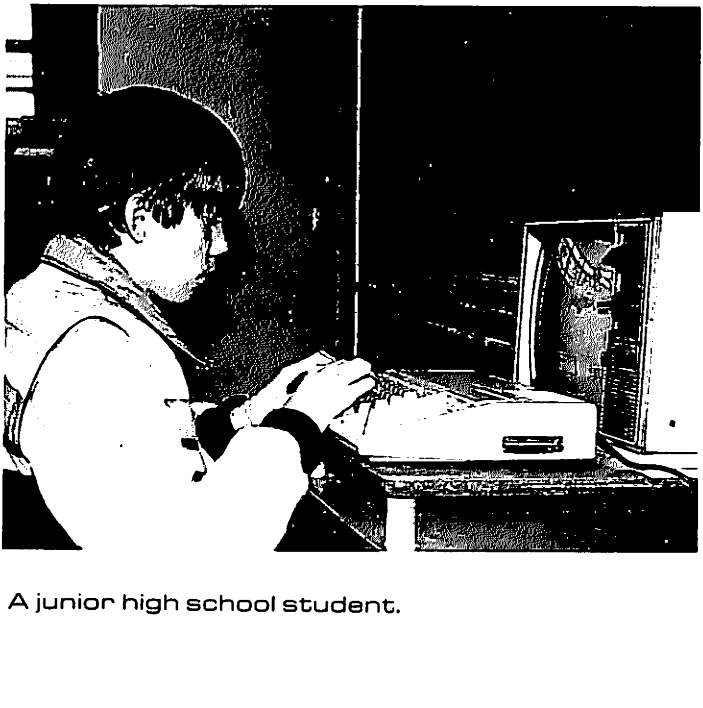
Computer Activities. Students are studying concepts in math, such as variables, that make programming in BASIC understandable. Coordination has improved, which will make typing on a word processor more productive. Drill games are still appropriate.

Expectations for High School

Abilities.

- Reading. Most students don't take a reading class. Instead, they read a lot in content classes, such as literature and history.

FIGURE 2-6



A junior high school student.

- Math. Some students continue in basic math classes. Others take specific math classes, such as algebra and geometry. Some continue into trigonometry and even calculus.
- Language. The focus remains on communication and expression, both written and spoken. Students are preparing for life on their own, whether getting a job or attending college. The ability to fill out a job application is important. Research papers are stressed. Grammar continues to be studied. Students also are studying foreign languages.

Attention Span. Attention span is not a significant consideration for most students.

Interests and Activities. Sports, social events, and jobs compete with homework time. Students are more and more independent.

Computer Activities. Some students are already learning to program, whether in a class or on their own. Word processing is a good

FIGURE 2-7

High school students.

job skill and also is useful for doing essays. Drill remains important for remediation in basic skills.

THINGS TO ENCOURAGE

The first thing to encourage is patience. Working with young people is sometimes frustrating. Young people think it is frustrating to work with adults, too. Remember that learning any new skill is sometimes awkward, and the computer is not an exception. The computer is so much fun for most young people to use that it's easy to forget that there are aspects of computer use that are difficult for them to learn.

The next thing to encourage is balance. To grow properly, a young person needs varied activities. A computer can be captivating. Some people spend hours in front of the computer. As with any other skill, computer skill is gained by regular practice. Computer skill, however, shouldn't take the place of other more important skills.

If you're working with your young person as a parent, then your child has already had a full day of school. Make sure that he gets enough exercise. Younger children, especially, need physical play to help develop the coordination required to learn to read.

Social development is another area to encourage. A computer is no substitute for a friend.

It's also important to have balance in computer activities. Computer games are fun and sometimes educational, but there are other ways to use a computer. We as humans need to tell the computer what to do, rather than letting the computer tell us what to do.

One of the enjoyable features of computer programming is experimentation. When you write a program, there is no one way to accomplish your goal. Many times you'll make a change to a program and not know exactly what effect the change will have when you run the program. Programs are easier to write when you use good programming style, which will be discussed in a later chapter, but the element of trial and error still makes programming exciting.

THINGS TO DISCOURAGE

We adults were once children and yet we forget many things about being children. One of the areas where this comes out is in introducing the computer to children. Most of us know how to type or, at least, to find typewriter keys in a reasonable amount of time. For a child, finding the proper computer keys can be time consuming. It's important to avoid making unrealistic demands on children using computers.

Most children enjoy using the computer, but some are frustrated by it. They lack coordination and have a short attention span.

I would strongly discourage pushing and impatience. It is certainly important that young people learn something about computers, but if you frustrate them when they're young, they may not want to use computers when they are old enough to use them more effectively.

Another thing to discourage is spending too much time at the computer. First, it can cause eye strain. Children spend all day in school and spend a lot of time reading. Some time at the computer is okay, but don't overdo it. Second, some students will allow the computer to take the place of exercise and friends.

chapter 3
USING COMMERCIAL SOFTWARE

One of the main strengths of a computer is its versatility. When you use a computer, you're using a machine of many uses. For instance, a computer can teach you new skills. You may learn skills you thought were way beyond you. As you proceed through the computerized lesson, you're amazed at how the computer converses with you by name. When you're having trouble with new information, the computer slows the pace and offers you hints. As your understanding increases, the computer presents material faster to match your new understanding. At points throughout the lesson, you're given questions to test your understanding of the material. As you finish the lesson, you're given a score.

If you've learned a skill, but need practice, the computer can help you there, too. It can present you with practice skills in an interesting format that makes you want to keep on trying. The computer seems to know just what level you're on, and makes it fun to practice.

Wouldn't it be fun to travel cross country to Oregon in a covered wagon? It sound impossible, but a computer can make the trip seem real. Situations are presented that test your decisionmaking skills and make you feel like you're really on the trail.

Maybe you have a long report to write. Editing is such a pain. It

FIGURE 3-1

would be so nice to be able to correct mistakes without having to retype a whole page. Oops, that paragraph is in the wrong place. Again, the computer comes to your aid. It is possible to correct mistakes on the computer screen as well as shift paragraphs.

The computer can accomplish these and many other tasks. It does not, however, do these tasks by itself. The missing ingredient is software. Software loaded into your TI via disk, cassette, or the keyboard turns the computer into a powerful tool.

If you've purchased a TI or looked into buying a TI, you'll be overwhelmed with the variety of software available for the TI. You can see products in computer stores, read computer magazines, or find yourself on new mailing lists.

A knowledge of software will help you come to grips with the problem of selecting it. The first section of this chapter discusses the various types of educational software. Each kind has strengths and weaknesses that will be examined.

As the types of software are discussed, some particular programs will be mentioned. They are good programs that the authors have used or seen demonstrated. There are other good programs available, but it is impossible to try all of them. Appendix E includes a list of recommended software.

The next section deals with the qualities of good educational software. To select good software, you have to know what to look for.

Finding software can be a very trying experience. The next section examines some software frustrations. On the basis of the preceding sections, the book makes some suggestions on selecting software.

TYPES OF SOFTWARE

This book discusses five types of educational software. These are tutorial, drill, simulation, game, and productivity software.

Tutorial

A tutorial teaches you new skills. The computer acts as a teacher and leads you through material step by step. Periodically the computer poses questions to see if you understand the concept presented. Ideally, the computer will branch to different parts of the program depending on your skill level. At the end of the program, a post-test should be presented so you can see how much you learned.

One example of a tutorial is TOUCH TYPING TUTOR by Texas Instruments. This program leads you finger-by-finger through a typing course. It can keep you posted on your speed, and lets you know if you're ready to progress to the next level or if you need more practice.

The advantage of a tutorial is that it can give individualized instruction to a student. In a class, a tutorial would be particularly useful in instructing a student who has trouble understanding a subject, or a student who needs more difficult material. For a parent, the tutorial would be helpful to provide help for a young person in a particular subject.

A problem with the tutorial is that in teaching a new skill, the tutorial cannot anticipate all the areas where a student will need help. A good teacher is sensitive to the student and can adjust the presentation accordingly. A good tutorial will adjust the presentation to the ability of the student but is not sensitive to the feelings of the student.

Another problem with the tutorial is that it uses a lot of computer time. If the tutorial takes 15 minutes for each student to complete, and the class has 30 students, it would take 7-1/2 hours for each student in the class to finish. Of course, if you're a parent working with one person, then the time factor may not apply.

Drill

The majority of educational computer programs are for drill. After you've learned a skill, drill and practice programs help you to review, maintain, or improve your skill level. A noncomputer example of drill is flash cards. Some computer drill programs cause the computer to act as electronic flashcards.

Drill is helpful for improving a skill, but should have certain qualities. First, it should be interesting. Computer arcade games are popular because they are fun. Drill and practice programs can be both enjoyable and educational.

Next, drill should be varied. Different situations should be provided for practicing the skill.

To be useful to a variety of persons, the practice should be individualized. Different skill levels should be available. The skill levels could be chosen by the machine or the user.

An example of a drill and practice program for the TI is NUMBER MAGIC by Texas Instruments. This cartridge will help a child practice basic math facts. The child can choose the difficulty and whether or not the exercises will be timed.

■ Simulation

A simulation is a re-creation of an event. The program CRIME AND PUNISHMENT, available from Decision Making Systems, Ltd., is a simulation of a courtroom situation. The user is given various bits of information pertaining to a crime and the accused. He then must determine the guilt or innocence of the accused, and the sentence, if the accused is guilty.

Simulations are particularly useful for certain kinds of events. Some are dangerous, such as a simulation of a nuclear reactor. Other simulations recreate events that are costly, time consuming, or even impossible.

■ Games

Tutorials, drills, and simulations can use a game format. Other games difficult to categorize stress logical thinking and problemsolving.

Adventure games available from various companies fit this category. These games, more often than not, use text only and no graphics. Users are placed in an imaginary environment and are required to plan their moves carefully in order to solve the mystery or adventure.

■ Productivity

Productivity software helps you work more efficiently or creatively. Word processing software, which will be discussed in Chapter 7, makes it easy for you to enter text into the computer. Then you can correct mistakes at the computer and have the text printed on a computer printer.

If you're a teacher, you might be able to use software to keep grades. (Try it out before you buy it to make sure it does what you want.)

CHARACTERISTICS OF GOOD EDUCATIONAL SOFTWARE

Unfortunately, not all computer software is of high quality. The book has examined some particular types of software. This section looks at educational software in general and gives some criteria for evaluating software.

The evaluation of educational software is an important issue. Since the software makes the computer function, it is actually more important than the computer itself. If we're to use the computer wisely to educate, it's imperative that we have good software.

The problem is that computers haven't been used in schools long enough and widely enough to accumulate a large body of knowledge about appropriate computer use. In another 10 years, some of the questions about computers and software will have been answered. Most of us using computers will not wait for the answers to be worked out, however.

Even though the software evaluation issue is a new one, we're not completely in the dark. Knowledge gained in other instructional areas can be helpful. For instance, think of some of the concerns you would have choosing a textbook.

You would want the text to be educationally sound. It should be as interesting as possible. You would want it to be easy to understand and to have an attractive format.

A computer program has some of the features of a textbook. It usually involves reading. Sometimes it has pictures (graphics). Usually there are questions to answer.

The computer, however, is much more powerful than the book, in several areas. A computer is capable of interacting with a student and presenting material based on student response. For instance, in a math drill, the computer can present easier problems if the student seems to be having trouble. For the brighter student, the computer can present more difficult problems.

When evaluating computer software, it's important to keep some things in mind. First, past experience with instructional materials and methods can help guide you in finding good educational software. Don't feel that since you have little computer background that you'll know nothing about choosing computer software.

Don't let your past experience, however, limit the way you think a computer can be used in education. The computer is not an electronic workbook. The computer is a powerful educational tool, whose potential has barely been tapped.

Second, don't expect software to take the place of a teacher. Even though excellent tutorials exist that can teach concepts and skills, a teacher is usually superior at relating to the individual young person, and the teacher has feelings which a computer never will have.

Evaluation forms exist that can help you choose software. Some evaluation form sources are listed in Appendix D. No matter which standard you judge software by, however, you won't find the perfect program. There are several reasons for this.

First, people disagree on qualities of a good program. Second, people believe in different theories. Third, software producers have different strengths. Some are excellent programmers. Others are strong on theory.

You'll have to be the final judge of a good computer program. As you gain experience using software, you'll become more sophisticated in evaluating it. You'll learn what is most important to you and what software faults you can accept.

In general, educational software has a few specific qualities. The most important quality educational software can have is to be educationally sound. Next, it should be easy to use. You want software that is interesting and that uses the computer well. Finally, you want understandable directions to come with the software.

■ Educationally Sound

Software that is educationally sound teaches something important to learn. It should have clear objectives and should meet those objectives.

It's important that the program is understandable for the intended user. Directions and explanations should be clear. The reading level and vocabulary should match that of the student's age level. Provisions should be made to match the instructional content to the ability of the individual student.

The software should be accurate. The facts should be correct. Watch out for errors in grammar and punctuation.

■ Easy to Use

A computer program should be easy to use even for the person with no computer background. Information on the screen should be formatted so that it's easy to read. When a response is required, the program should make it obvious what type of response is required.

When you read a book, you don't have to read the pages in order. If you're working on exercises, you sometimes have to review material or reread the directions. When you're through, you simply close the book.

These features can be built into a computer program. The student should be able to backtrack through the program and review materials or receive help when necessary. The speed of the program should be determined by the student and there should be opportunities to exit the program, short of turning off the computer.

■ Interesting

If a program is interesting, it can be used to teach or review skills that would be difficult to do using different programs. The computer has unique abilities that are strong motivators. These features will be discussed in the next section.

The best test of whether a computer program is interesting is to watch young people use it. If they enjoy a program, they will keep at it, even if not required to. Later, they will come back to it and sometimes tell friends about it.

■ Uses Computer Well

Computer time is valuable. If you're a teacher with one computer in your school or even in your classroom, you know that you could have students using the computer every mintue of the school day, including breaks. If you're using the computer with your son or daughter, you don't want them to spend too much time at the computer.

The question that has to be asked is whether a program is a good use of computer time. Could its goals be accomplished better using pencil and paper or other methods?

The computer's unique abilities can motivate a person. Young people enjoy the fact that a computer can remember their name. They also like it when the computer presents material at a level that they can understand.

Since a computer works so fast, it can keep track of student progress and give instant feedback. The feedback can include a score and even elapsed time.

FIGURE 3-2

Another strength of the computer is its ability to use color graphics and sound. Watch out for sound, however, in a classroom. You might want to turn down the volume.

Finally, the computer can be programmed to present information in a random order.

Documentation

The written material that comes with a computer program should be clear and easy to understand. It should include directions for the student and for the teacher.

SOFTWARE FRUSTRATIONS

Searching for good quality educational software will probably be very challenging. It seems that it would be easy to find software for a particular application. Certain factors, however, will complicate your software search.

Availability. It's easy to buy a TI computer. You go to a department store, try it out, and take it home. Unfortunately, the same store usually stocks very little educational software. Most of it has to be ordered.

But from whom do you order it? Some is found in this catalog, some in that, and some has to be ordered from the publisher. Appendix A includes some software recommendations.

Lack of Information. Now that you've located sources of software, the next problem is a lack of information. Reading a description of a program in a catalog does not give you the information you need. For instance, you'll want to know what level students would use it. You will want to know if the software is interesting and performs as advertised. You really need to try out the software.

Fortunately, some software distributors sell software on a trial basis. If the software is unsuitable for your needs, you may send it back. Often, however, you do not get enough information to know whether you even want to try it out.

Machine Compatibility. Different models of TI computers will run different programs. A program written in EXTENDED BASIC won't run without the EXTENDED BASIC cartridge. Some programs will require a 32K memory expansion unit, and others a disk drive and controller. Usually, a program description will include language and memory requirements, but it's easy to get confused if you're a computer beginner.

Copy Protection. Many software producers prevent their software from being copied, modified, or even looked at. This is called copy protection. There is a reason for copy protection. Unprotected computer programs are easy to copy. Many people would copy a program from someone else rather than buy it.

Unfortunately, copy protection causes problems for the software user. One problem is backup. It can be very frustrating if your disk is damaged and you have no backup. Software producers, however, are required to include a backup disk or sell one at a reduced price.

Another problem is that you cannot modify a protected disk. With an unprotected disk, you can repair program bugs and change the program to fit your needs.

SOFTWARE ADVICE

Don't be disheartened by software frustrations. Microcomputing is still in its infancy. New programs are introduced weekly, and program quality is rising. Competition will force inferior programs off the market. As more schools and individuals buy computers, the software will become more readily available.

Your knowledge about software will increase as you gain experience with the computer. As you read this book and work the programming exercises, you'll become more sophisticated at choosing software. Other ways to gain computer knowledge are to take computer classes, read other computer books and magazines, and join a computer club.

Experience also will aid you in finding programs compatible with your TI. You'll learn what to look for. Here are some specific suggestions that will help you to obtain good software.

■ Consider Your Needs

First, consider your needs. If you're an elementary school teacher with one computer in your class one day a week, you won't need very much software. Something to introduce students to the computer will work fine.

If you're a parent who wants to use the computer to drill your child on math, you won'tneed software designed to drill and keep track of 30 students.

On the other hand, if you want to use computers to teach math to 30 students and track their progress, you won't be satisfied with software designed for one student.

■ Explore Public Domain Software

Software not copyrighted is called public domain software. It can be copied freely. One source of public domain software is a local TI computer club. The computer club receives software from other clubs. Some of the donated software is educational.

Another place to look for free software is your local school system. Many schools use TI computers and have access to free or nearly free software.

■ Learn More About Commercial Software

Talk to other TI computer owners and see what advice they have. Check with the school system again. Find out if they're using computers and if so, what software they're using. If you're a parent, talk to your child's teacher to find some ideas on what skills your child needs to develop.

Another good idea is to check with a local computer store. Some stores have educational specialists or some software you can try out. If they have software to try, have a young person try it out.

If you order software from a catalog, make sure you can return the software if it doesn't meet your needs.

chapter 4 DOING PROGRAMMING

This is potentially the most important chapter of the book. Learning to program a computer will bring you many benefits.

IMPORTANCE OF PROGRAMMING

Learning to program will teach you a new skill. It's a skill that could help you find a job. In a few years, the majority of jobs will somehow relate to a computer. Some of the jobs will involve a computer directly. Others will use computer information.

Most of the jobs that deal with computers fall into four categories: computer programmer, operator, technician, and systems analyst. Studying programming will help a person considering any of these jobs.

Obviously, a programmer would profit from any previous programming experience. There are many different programming languages. Some will be discussed in a later section. Usually, learning one programming language will make it easier to learn other languages.

The computer operator is a person who runs the computer. Usually, the operator works with a mainframe or minicomputer. These are computers that are larger than the TI. The duties of the operator include loading programs and data and collecting the output. The operator also powers up and shuts down the computer. A knowledge of programming will help the operator gain insight into possible computer problems.

The technician maintains and repairs the computer system. A knowledge of programming will assist the technician in diagnosing computer problems.

The systems analyst uses specifications given by a customer and chooses the appropriate computer system. To become a systems analyst, you start out as a programmer.

Besides the traditional computer jobs, new jobs are opening up. Jobs that don't exist will be created in years to come. A knowledge of computer programming, no matter what the language, can help you prepare for the future job market.

Even if you have no plans to enter a computer career, computer programming can help you. To write computer programs, you'll need to be well organized and approach the problem logically and systematically.

Computer programming can help you to become more orderly and logical in your thinking. This orderly thinking will carry into other areas of your life, such as writing a paper or figuring out why your car isn't functioning properly.

Besides learning a valuable skill, you'll gain a feeling of mastery over the computer. Many people spend a lot of time playing computer games. This is fine if you don't overdo it. When you're playing a computer game, however, you're not in control of the computer. In a sense, the computer is controlling you, instead of you controlling it. Programming the computer can contribute to self-confidence. It's a real accomplishment to instruct a computer and have it follow your orders.

PROGRAMMING LANGUAGES

Programming the computer consists of giving the computer directions in a language it can understand. The language of the computer is called machine language. Machine language consists of ones and zeros, the digits of the binary, or base two, number system.

The TI understands another form of machine language. The instructions are the same, but they're expressed in the hexadecimal number system. The hexadecimal number system, hex for short, uses base 16. The digits are 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F. It looks strange to see a letter standing for a numeral, but you get used to it.

No matter what number system is used, instructions made up of numbers make no sense to most humans. To make programming easier to understand, simple programming languages were developed.

- Assembler. Assembler, or Assembly language, is one step up from machine language. It use three- or four-letter operation codes for each machine language instruction. As an example, the operation code LDA stands for load accumulator. Assembler is a big improvement over machine language, but it's still very difficult for most people to learn.
- FORTRAN. FORTRAN, which stands for Formula Translator, was the first computer language that used easily understood instructions. It was designed for tasks that require complicated numeric computations, such as those in science and engineering.
- COBOL. A need then arose for a computer language that would meet the needs of business. COBOL, Common Business Oriented Language, was developed. The language is stated in less mathematical terms and is oriented toward producing formatted reports needed in business.
- Pascal. If you plan to study much programming at a college or university, you'll probably study Pascal. It's a logical, well-structured language that will teach you good programming habits, no matter what other languages you study. Pascal is becoming an increasingly important programming language.
- BASIC. Beginners All Purpose Symbolic Instruction Code was developed to meet the need of beginning computer students. It was designed with the idea of using a few easily understood instructions. BASIC is the most popular language on microcomputers.
- LOGO. For young children and others, BASIC is too difficult to understand as a beginning computer language. LOGO was developed to teach programming to children. One feature of LOGO is Turtle Graphics. Simple commands direct a turtle around the computer screen, which draws pictures as is goes.

  LOGO is a fairly new language for the microcomputer. It's a good language for introducing children to programming.
- PILOT. Some people need to prepare computer-assisted instruc-

tion, but don't have the desire or time to learn a programming language like BASIC. PILOT is an example of an "author" language. It allows a teacher to make up a computer lesson with a minimum of programming.

CHOICE OF A PROGRAMMING LANGUAGE

Each of the programming languages listed has advantages. I have chosen two languages to use in this book. They are BASIC and LOGO. Both languages have advantages for learning, which will be discussed.

■ LOGO

LOGO, for microcomputers, is a new language. It was developed to teach children to program a computer. Its appeal is that you can write simple but interesting programs.

LOGO uses an imaginary turtle that leaves a trail on the computer screen. Simple instructions, such as FORWARD, LEFT, and RIGHT, move the turtle all over the screen. Just by using those three commands, a child can explore mathematics and even art.

LOGO is not just for children, however. People of all ages can use LOGO to prepare for other programming languages and to explore concepts of programming.

To write a computer program, you first break the task down into smaller units. Your main program should link the smaller units (or procedures). The smaller units then can be written separately. After the program is written or while the program is being written, you have to test the program and debug (remove errors from) it.

The LOGO language is set up to encourage orderly development of programs. You can program in units called procedures. Procedures don't need line numbers, as in BASIC, and don't need to have variable names that match the variable names in the main program. Thus it's easy to test procedures independently of the main program. Furthermore, when you make an error in LOGO, the error messages are more descriptive of the problem than in BASIC.

If you choose to learn LOGO, you'll need the proper hardware and software. Besides the console and monitor, you'll need the peripheral expansion system with the memory expansion card. To store programs, a program recorder or disk memory drive with a disk controller card is required. The actual LOGO language is contained in the software package TI-LOGO or TI-LOGO II.

This book contains five LOGO programming worksheets. The purpose of these worksheets is to give you an introduction to the LOGO language. They should give you a glimpse of the simplicity and power of LOGO. To become proficient in LOGO, you'll need to study and practice the examples in the TI-LOGO manual.

The first worksheet is appropriate for an elementary school student who can read and follow directions. The other worksheets are written for fifth graders and older students, as well as adults. Even though LOGO was written for children, many of the powerful ideas of LOGO are better understood and appreciated by older students and adults.

If you have doubts about the usefulness of these worksheets with a certain age group, try them out.

■ BASIC

The biggest advantage of BASIC is its popularity. Most computers made can run a version of BASIC. If you learn BASIC on the TI, you'll be able, with slight changes, to program other computers.

BASIC is not exactly the same on all computers, however. For instance, graphics (commands that draw pictures on the screen) are not at all the same on various computers. Most of the BASIC instructions, however, are similar.

Another consideration for choosing BASIC is cost. The inexpensive unexpanded TI will run BASIC. You don't need a disk drive, memory expansion, or even a module (unless you want to learn EXTENDED BASIC).

If you want to write long programs, or programs that need to execute quickly, BASIC (or Assembler in some cases) probably will be your choice. The BASIC language itself, compared with LOGO, takes up little room in memory and executes instructions quickly.

The programming examples in the next chapter are written in BASIC, although many of the programming ideas apply equally well to LOGO.

This book contains 12 BASIC programming worksheets to help you learn BASIC. These worksheets, along with the TI programming manuals and a lot of practice, will give you a good foundation in the BASIC language. From these worksheets, you can then go into writing software in the next chapter.

If you're working with junior high school or older students, these worksheets should be within the ability level of most students. Be prepared to give more help, however, with advanced topics, such as arrays.

Fifth and sixth grade students should be able to handle some of the worksheets, depending on their abilities.

If you have any doubts, test the worksheets and see how the students do. If you're working with a gifted class, of course, they'll be able to understand more difficult concepts.

LEARNING TO PROGRAM

Teaching Yourself

You can teach yourself to program. Your progress will be determined by several factors: desire, practice, and ability.

Programming is difficult at times. To succeed, you'll need a strong desire to program. Learning with a friend is helpful. When you get discouraged, you'll have someone to encourage you. If your friend is having trouble with something you understand, you can help your friend.

Even if you want to learn to program, you'll have to practice. Programming is like a foreign language. To develop fluency, you have to speak it (actually, type it) often. If you have a certain amount of time each week, it's better to break the time up into blocks, rather than have one long session. Four 15-minute sessions are better than one 60-minute session.

FIGURE 4-1

You mean you can't learn to ride a bike by just reading instructions?

If you have the desire to learn programming and take the time to practice, then ability also will help you. You don't have to be a talented programmer, however, to write programs that can be used by others.

To get maximum benefit from this book, work the programming exercises on the computer and answer the written questions. It's often tempting to read the exercises only, instead of trying them on the computer, and skip the questions. You can learn some programming from just reading, but you won't progress very fast, and you won't get the feel of it. It's a little like trying to learn to ride a bike by reading the instructions, but not practicing.

As you learn new programming skills, don't be afraid to experiment. You don't need to worry about hurting the computer. Sometimes, you'll have questions about what will work on the computer. The best way to find out is to turn on the computer and see what happens.

■ Teaching Others

If you're teaching others to program, carefully consider the abilities of your students. (If you're a parent teaching your young person, you're a teacher!) Some students want to program the computer and are willing to put in the time practicing, but don't have the maturity for certain computer concepts.

An example of a difficult programming concept is the variable. The statement A = 5 does not make sense for most elementary students. By junior high age, most students are ready to tackle variables. Of course, students vary widely in their ability to handle abstract concepts, such as variables.

Another area where elementary students run into difficulty is in typing. They do fine if the assignment isn't too long.

No matter what ability level you're working with, don't push too hard. It may look to you that students aren't trying when they really do not understand the lesson.

■ Difficulties

Learning to program is difficult, especially if you're teaching yourself. Exercises are hard to understand. Programs often won't run. When they do run, it seems that you put in so much work for so little results.

Sometimes programming does not make sense. Variables can be difficult to understand. The idea of loops in a program is often confusing. At times like this, it's good to find someone who can help you. Usually, people who understand programming are happy to help others.

If you have no one to help you, it may help to keep going back over the material. Maybe, someone who does not program can even look over the material and help you. Sometimes it's easier for someone else to spot the problem.

It's frustrating to have a computer program that does not run. You have removed the serious bugs from the program and cannot see any reason that the program does not run. You have gone over the program line by line, and still it does nothing.

Again, it helps to show the program to someone else. Sometimes, the problem is very simple. Maybe one variable name is misspelled. You may have typed a semicolon where you need a colon. A computer isn't smart. It has to have its instructions spelled out exactly correctly.

Finally, you get the program to run. The next problem may be frustration that you wasted so much time on a program that prints out, "Hi, I am a computer." You wonder sometimes what the point is of putting in so much effort for so little return.

If you'll be patient and persistent, you'll notice your computer programs growing in sophistication and usefulness. Remember, at first it was difficult to ride a bike. When I learned to ride a bike, I launched myself from the side of a porch, because the bicycle was too big for me. When I was ready to stop, I either made it back to the porch or crashed. Eventually, I learned to ride the bike, and you'll learn to program if you keep at it.

DISPLAYING INFORMATION

Before you start pressing buttons, take a moment to examine the keyboard. The keyboard looks similar to a typewriter keyboard. If you look closely, however, you'll notice some differences. A few of the symbol keys are in different positions, and there are a few new symbols. You'll also notice that a few keys make no sense at all.

We will discuss the meaning of some of the special keys on the TI. As the keys are mentioned, you can locate them on your TI.

ENTER. This is like the carriage return key on an electric typewriter. It tells the computer that you have replied to a question or reached the end of a line. Unlike a typewriter, the computer line can be up to 239 characters wide. At first, you may have trouble remembering to press the ENTER key at the end of each line.

Space bar. It's used to put spaces between letters or words, like on a typewriter.

SHIFT. This works the same as a typewriter SHIFT key. Pressing the SHIFT key with a letter key will produce a capital letter. Some of the keys, such as the numbers, however, will be different if the SHIFT key is pressed. Holding down the SHIFT and the 8 keys at the same time will produce a *. If a key has two symbols, the symbol on the upper part of the key appears using the SHIFT key.

CAP LOCK. If you press the CAP LOCK key, it tells the computer to print only the uppercase letters when a letter key is pressed. Some of the keys, such as the numbers, however, work the same as they did before you pressed the CAP LOCK key, printing the lower symbol or the upper symbol, if pressed with the SHIFT key. If the computer is printing all capital letters, press the CAP LOCK key and then release it. This will return the keyboard to normal upper-lowercase operation.

FCTN. This is the function key, and is used with other keys, like the SHIFT key. If you press FCTN and 4, this is called "function clear." This commonly is known as the BREAK sequence and will stop program execution.

Number ∅ and Letter O. On a typewriter, you can use the number 0 for the letter O. The computer won't allow you to do this. Notice that the number 0 has a slash through it.

Number 1 and Letter l. The computer knows the difference.

+, −, *, /. These are signs for addition, subtraction, multiplication, and division.

← and →. These keys, used with the FCTN key, allow you to move the CURSOR left or right within a line.

chapter 5 PROGRAMMING IN BASIC

LESSON FORMAT

Each lesson will be divided into three parts. The first part will be the introduction. It will give you helpful information to help understand the lesson.

The second part of the lesson is the computer lesson. There will be a worksheet for you to complete. The worksheet will be divided into two parts. The left side of the sheet will contain instructions you're to type into the computer. The right side will have questions for you to answer.

The last part of the lesson is the discussion section. It will start off with the answers to the worksheet. Then it will continue with helpful facts and rules pertaining to the TI version of BASIC.

LESSON 1

Now we're ready to get started. One of the first things that will strike you about the computer is how exacting it is. It will do exactly what you tell it to. Just make sure you tell it what you mean.

One of the first problems you'll have is typing in information from the worksheet. The worksheet is divided in half. You'll type the part on the left side of the sheet. The first line says

```
PRINT "HI" !E!
```

On the computer, you'll type

```
P R I N T " H I "
```

and then you'll press the ENTER key, as shown by !E! on the worksheet.

Be careful to put quotes in the right places. To type a quote, remember to type FCTN. Hold it down and press the "P" key.

If you make a mistake in typing, you can use the FCTN-S combination to move back over the mistake and type the correct letter. If you need to remove a letter, position the cursor over the mistake, using FCTN-S to move left or FCTN-D to move right. When the cursor is over the letter, press the FCTN key with the "1" key to DELete the character. You also can INSert characters. To INSert, move the CURSOR to the right place, press the FCTN key, and, while holding it, press the "2." Anything you type now will be INSerted to the LEFT of the CURSOR until you press ENTER, move the cursor with the arrow keys, or use one of the other FCTN combinations. If you notice a mistake after you press ENTER, just type "EDIT" and the line number, and press ENTER. The line will be listed with the cursor positioned over the first character. You may make your corrections using the same methods. When you're finished, press ENTER to store the revised line.

If you get stuck, look ahead to the discussion section. It won't be cheating to look at the answers. Sometimes you may need to go over a worksheet several times before you understand it.

Now it's time to turn on the computer and begin! Here are the steps.

1. Turn on the TV.
2. Turn on the expansion interface, if you have one.
3. Turn on the computer.
4. Press any key to get the menu.
5. Press the "1" key for TI BASIC.

□ Discussion

You are now on your way to learning programming. It's just a matter of

WORKSHEET 1

TYPE THIS ON THE COMPUTER	THEN ANSWER THESE QUESTIONS
PRINT "HI" !E! PRINT "HOW ARE YOU?" !E! PRINT "5 + 3" !E! PRINT 5 + 3 !E! PRINT "5 + 3 =";5 + 3 !E!	What did the computer print? 1. ____ 2. ____ 3. ____ 4. ____ 5. ____
1Ø PRINT "I" !E! 2Ø PRINT "AM" !E! 3Ø PRINT "GREAT" !E!	Now for a program. Do not forget to type line numbers. 6. Did the computer repeat what you typed? ____
CALL CLEAR !E! RUN !E! LIST !E!	7. What happened? ____ 8. What was printed? ____ ____ 9. What was printed? ____ ____
NEW !E! CALL CLEAR !E! RUN !E! LIST !E! CALL CLEAR !E!	10. Was the message printed? ____ 11. Did the computer list the program? ____

following directions and practicing. In this section of the lesson, the answers to the worksheet questions will be given and discussed.

Questions 1 to 5 deal with the computer in what is called the immediate mode. In the immediate mode, the computer performs some action as soon as you type a line and press the ENTER key.

1. HI
2. HOW ARE YOU?
3. 5+3

In the first three examples, the computer printed back what you typed in. Material enclosed in quotation marks will be printed as you type it.

4. 8

5. 5+3=8

In number 4, the computer doesn't print what was typed because there no quote marks. The computer printed the value of 5+3, which is 8. In example 5, the computer printed 5+3=, which was enclosed in quotes, and then printed the value of 5+3. The semicolon separated the two parts of the statement. In future lessons, we will study the semicolon.

In the next section, we get into the deferred or program mode. It's called the deferred mode because the computer doesn't perform an action until all the statements are typed into the computer.

6. No

7. The screen went blank.

When the screen went blank, did you think that you had hit a wrong key? Don't worry. The program is still in the computer memory.

8.
```
I
AM
GREAT
```

9.
```
10 PRINT "I"
20 PRINT "AM"
30 PRINT "GREAT"
```

The words, CALL CLEAR, RUN, and LIST are examples of commands. Commands are words that cause an action after the ENTER key is pressed. CALL CLEAR clears the screen. RUN caused the program to execute and print "I AM GREAT." LIST brought a copy of the program from the memory to the screen.

10. No

11. No

NEW is another command. It clears the memory for a new program and also clears the screen. Make sure that you understand the difference between the commands NEW and CALL CLEAR. NEW clears the memory and the screen. CALL CLEAR clears the screen but not the memory. You can type in a program and type CALL CLEAR to clear the screen. The program is not lost, however. To see it again, type LIST. If you want the program to execute, type RUN.

LESSON 2

In Lesson 1, you used the PRINT statement for various cases. In this lesson, other features of the PRINT statement will be used. Before you get started, however, you should notice a few things.

First, don't forget to press the ENTER key at the end of each line. There will be no reminder on the worksheet itself.

Second, be sure to type the lines exactly as printed. Some of the lines end with commas or semicolons. It's important that you type them in, too.

WORKSHEET 2

TYPE THIS ON THE COMPUTER THEN	ANSWER THESE QUESTIONS

10 PRINT "THIS";
20 PRINT "IS";
30 PRINT "FUN"
RUN
	Don't forget the semicolon.
	1. What was printed? ____________
	________________

NEW CALL CLEAR 10 PRINT "THREE", 20 PRINT "BLIND", 30 PRINT "MICE" RUN	Clear the memory and screen. Notice the comma. 2. What was printed after you typed RUN? ____________
NEW 10 CALL CLEAR 20 PRINT TAB(10);"THIS IS A" 30 PRINT TAB(10);"TI-99/4A." RUN	This will also work in a program. 3. Where was the message printed? ________________
25 PRINT 35 PRINT " ":" ":" ":" ":" ":" ": " ":" ":" ":" ":" " RUN LIST NEW CALL CLEAR	Line numbers do not have to end in 0. 4. What is different? ____________ 5. Are the line numbers in order? ________________

This is an important lesson because it concerns the format of the printing on the screen. If you write your own programs, it's very important that the text on the screen be spaced so that it's easy to read.

□ Discussion

There are various ways to format the printing on the screen. The first way is with the semicolon.

1. THISISFUN

Usually, each statement is printed on a separate line. The semicolon, however, keeps the printing packed together on the same line.

2. THREE   BLIND
   MICE

The TI has two print zones. The comma causes each word to be printed in a separate zone.

3. The message was printed in the middle of the line.

If you've done much typing, you are probably familiar with setting tabs. The TI has a TAB statement. It's always used as part of a PRINT statement. Here is an example:

```
PRINT TAB(10); "HI"
```

4. The message moved up to the center of the screen.

All PRINT statements cause something to be printed at the bottom of the screen. Everything that is already on the screen gets moved up one line for each line that is printed at the bottom. The top line disappears. This is called scrolling. If you tell the computer to PRINT "", since there is nothing between the quotes, the computer prints nothing. This does, however, cause everything to "scroll" up one line. If you want to print several things on separate lines, you don't need a separate PRINT statement for each line. Separate each set of quotes and whatever is in them with a colon. Line 35 tells the computer to PRINT 11 lines of nothing, thus moving the message up 11 lines.

5. Yes, the line numbers are in order.

Notice that in most of the programs, line numbers increase by 10. The line numbers can be any numbers in the range of 0 to 32767. The reason for numbering by 10 is to leave room to insert statements between lines if necessary.

In the program, we have the lines:

```
20 PRINT TAB(10);"THIS IS A"
30 PRINT TAB(10);"TI-99/4A."
```

We wanted to insert the statement PRINT between lines 20 and 30, so we chose line number 25. When the program was listed, the line numbers were in order:

```
20 PRINT TAB(10);"THIS IS A"
25 PRINT
30 PRINT TAB(10);"TI-99/4A."
```

LESSON 3

For this lesson, you will use the immediate mode of the computer. When you type a line into the computer and press ENTER, you'll see the answer immediately.

The computer can be used as a calculator. It has an advantage over most calculators. The results of your calculations will remain on the screen, as long as there are not too many calculations.

Notice that the computer uses special symbols for multiplication and division. The symbol for multiplication is *. The "×" sign is not used, because the computer would get confused with the letter x. Remember, the computer is not smart enough to know the difference. Humans have important advantages over computers.

The computer uses the "/" for division. If you consider the fraction 1/4, the fraction means 1 divided by 4.

Another difference you'll notice between the computer and traditional math is that you don't use commas when entering numbers. Consider the numeral 123,456. To you, the numeral is one hundred twenty-three thousand, four hundred fifty-six. To the computer, it's two numerals, one hundred twenty-three and four hundred fifty-six.

□ Discussion

The first problems are straightforward calculations.

1. 579

2. 333

WORKSHEET 3

TYPE THIS ON THE COMPUTER THEN	ANSWER THESE QUESTIONS
PRINT 123 + 456 PRINT 456 – 123 PRINT 123 * 456 PRINT 456 / 123	What did the computer print? 1. ________________ 2. ________________ 3. ________________ 4. ________________
PRINT 1 + 2 * 3 + 4 PRINT (1 + 2) * (3 + 4)	5. Are you surprised? ________ 6. Did the computer add or multiply first? ________________ 7. Did the computer add or multiply first? ________________
PRINT 999999 * 999999	8. What is the answer? ________ The numeral is written in scientific notation.
PRINT 999999999999999 – 999999999999998 CALL CLEAR	Do not use commas in the number. 9. What is the computer's answer? ________________ 10. Is the computer correct? ______

3. 56088

4. 3.70731708

Notice, in number 4, that the computer gave the answer correct to 9 digits. Another way to say that is: The answer has 9 significant digits.

5. You may be surprised and then again you may not.

6. The computer multiplied first.

When you have a chain of calculations, a calculator will simply perform the calculations as the numbers are entered. A computer, however, follows rules when processing a number of calculations in a row. Certain operations will be performed first and others will wait. The order in which operations are performed is called precedence.

As you saw in number 6, multiplication is performed before addition. A simple phrase can remind you which operations have first priority:

My	multiplication
Dear	division
Aunt	addition
Sally	subtraction

If the computer is to calculate a chain of calculations involving all the above operations, the computer will go through the calculations from left to right. It will first do all the multiplication and division problems. Next, it will do the addition and subtraction problems.

Multiplication and division have the same priority. Likewise, addition and subtraction operations have equal priority.

"My Dear Aunt Sally" summarizes the main rule of precedence. For complete rules, refer to the TI BASIC Programming Reference Manual.

7. The computer added first.

Sometimes, you'll want the computer to perform calculations in an order different from the rules of precedence. The way to do this is with parentheses. Operations enclosed in parentheses will be performed first. In the example,

```
PRINT (1+2) * (3+4)
```

the problems 1+2 and 3+4 were evaluated first. Next, the computer multiplied 3 and 7 to get the answer, 21.

8. 9.9998E+11

Numerals with more than 11 digits are expressed in scientific notation. The answer above is equivalent to 9.9998 × 10^11 or 999,980,000,000. The answer is not exact, because the TI's accuracy extends only to 11 digits.

9. 0

10. No, this is not correct.

Here is another case where the TI is not quite accurate.

LESSON 4

Variables give the computer power and personality. One example involves a program where the computer asks you for your name. Later in the program, the computer will address you by name.

It's a variable that makes this possible. When you type your name into the computer, your name is stored in memory locations in the computer. The variable name, for instance NAME$, points to the memory locations where your name is stored. Each time the statement PRINT NAME$ comes up in the program, the computer looks in the memory locations pointed to by NAME$ and prints your name.

□ Discussion

The first part of this lesson is concerned with numeric variables.

1. 1234 5678

WORKSHEET 4

TYPE THIS ON THE COMPUTER THEN	ANSWER THESE QUESTIONS
NEW 10 A = 1234 20 B = 5678 30 PRINT A, B RUN	1. What was printed? ____________
30 C = A + B 40 PRINT C RUN LIST	2. What was printed? ____________ 3. What numbers were added? _____ ____________________ 4. What happened to 30 PRINT A,B? ____________________
40 PRINT "THE ANSWER IS"; C RUN	Notice the semicolon. 5. What was printed? ____________ ____________________
NEW 10 A$ = "I AM" 20 B$ = "THE GREATEST" 30 PRINT A$ 40 PRINT B$ RUN	Clear the memory and screen. A$ is pronounced "A STRING" 6. What was printed? ____________ ____________________
30 C$ = A$ & B$ 40 PRINT C$ RUN NEW	7. What was printed ____________ ____________________

When the statement A = 1234 is made, the computer finds a location in memory and labels it A. For B = 5678, the computer finds a location and labels it B. The command PRINT A,B causes the computer to print the contents of the memory locations labeled A and B. Notice that the comma causes the numbers to be printed in separate print zones.

In some versions of BASIC, the word LET has to be used with the variable name, for example LET A = 1234. In TI BASIC, LET is optional.

2. 6912

3. 1234 and 5678

4. It was replaced by 30 C = A + B.

The equation C = A + B caused 1234 and 5678 to be added. At this point, the computer knew the answer, but we did not. To have the answer displayed, the command PRINT C is required.

When the statement 30 C = A + B was added to the program, the statement 30 PRINT A,B was erased from memory.

5. THE ANSWER IS 6912

An advantage that a computer has over a simple calculator is that the computer can print out results in a more understandable form. In number 5, the semicolon is used to separate the quotation and result sections of the PRINT STATEMENT. Did your answer have a space between the word "IS" and the numeral 6912? If not, make sure that there is a blank between the S and the second quotation mark in line 30.

The rest of the lesson is concerned with string variables. A$ is an example of a string variable. It can contain numbers as well as other characters.

6. I AM
   THE GREATEST

7. I AMTHE GREATEST

A$ and B$ point to memory locations where "I AM" and "THE GREATEST" are found.

Just as numerals have operations such as addition, so do strings. The operation in example 7 is called concatenation and uses the "+" sign. The equation C$=A$+B$ "adds" the message "I AM" to "THE GREATEST."

There are other string-related functions and operations. For more information, refer to page II-99 of the TI BASIC Programming Reference Manual.

LESSON 5

If you write much software, you'll use the INPUT and the CALL KEY statements a lot. When you request information from the user in a program, you usually use either INPUT or CALL KEY.

Before we actually get to the worksheet, let's review a few things. Here is a list of the main commands learned so far:

RUN	Start a program
LIST	Print a copy of the program
CALL CLEAR	Clear the screen
NEW	Erase the memory and the screen
FCTN-S	Move the CURSOR left
FCTN-D	Move the CURSOR right
FCTN-1	DELete a character
FCTN-2	INSert characters

Are you having trouble with errors? Here are some hints. If you make a mistake and you have not pushed ENTER, use the FCTN-1 keys. The real problem starts when you discover the mistake when you're on another line. Usually, the remedy is simple. Just retype the line.

Sometimes, however, you might make the mistake of typing a line number ending in the letter O instead of the number 0. If you mean to type 20 PRINT "HI", but type 2o PRINT "HI", the computer will list the statement,

```
2 O PRINT "HI"
```

Your hint that somthing is wrong is the space between the 2 and the O. The computer in interpreting the 2 as the line number and the letter as a command. To correct the mistake, first type 2 and push the ENTER key. That erases line 2. Next, retype line 20.

When you correct a mistake, the mistake usually remains on the screen. Don't worry. What you typed most recently is what stays in the computer memory. If you retype line 20, the old line 20 is erased from the computer memory.

Since you should now be familiar with the command RUN, it won't be included on the worksheet. The worksheet, is divided into sections by solid horizontal lines. When you have typed a program or added to a program and reach one of the lines, it's time to stop, type RUN, and see what happens.

From time to time, it's a good idea to type CALL CLEAR to clear the screen. Remember that CALL CLEAR does not clear the memory. You don't have to worry about losing your program.

When you type and RUN a program that contains an INPUT statement, you'll have to provide information to the computer. For

WORKSHEET 5

TYPE THIS ON THE COMPUTER THEN ANSWER THESE QUESTIONS

```
NEW
1Ø CALL CLEAR
2Ø REM ASK FOR DISTANCE
3Ø PRINT "HOW MANY MILES DID
   YOU TRAVEL?"
4Ø INPUT DIST
5Ø REM ASK FOR NUMBER OF
   GALLONS
6Ø PRINT "HOW MANY GALLONS
   OF FUEL WERE USED?"
7Ø INPUT GAL
8Ø REM CALCULATE AND PRINT
   FUEL MILEAGE
9Ø MIL = DIST / GAL
1ØØ PRINT "YOUR MILEAGE IS";
    MIL
11Ø PRINT "MILES PER
    GALLON."
```

Run this program. When asked for the number of miles, type 1ØØ. Type 5 for gallons of fuel.

1. What answer did the computer give? ______________________
____

Run this program several more times using different values.

2. When the computer asks for distance or gallons, push RETURN without giving a value. What happened? ______________________
____
3. When the computer asks for a value, type FOUR instead of 4, for one value, and TWO for the other. What answer does the computer give? ______________________
____

```
NEW
1Ø CALL CLEAR
2Ø REM ASK FOR NAME
3Ø INPUT "WHAT IS YOUR NAME?
   ": NAM$
4Ø REM PRINT A FRIENDLY
   MESSAGE
5Ø CALL CLEAR
6Ø PRINT "PLEASED TO MEET
   YOU "; NAM$
7Ø PRINT " ":" ":" ":" ":" ":" ":" ":" ":" "
```

4. After you give the computer your name, what does the computer print? ______________________
____
5. Now run the program again. For your name, type 1234. What does the computer print? ______________________
____
6. Does the computer know if you are using your real name? __________
7. Do the REM statements show up in the program when you run it? ____

instance, the computer may print "WHAT IS YOUR NAME?" You will then type your name into the computer and press ENTER.

If, in the worksheet, you get the statement "* INPUT ERROR" after you're asked for information, you'll have to enter a number and then press the ENTER key.

□ Discussion

The first program uses the INPUT statement with number variables. This program can be used to calculate fuel mileage.

1. YOUR MILEAGE IS 20 MILES PER GALLON.

This can be a useful program for someone who needs to calculate fuel mileage and knows little about the computer. When the program is run, the computer prompts the user to type in distance and number of gallons of fuel. Then the answer is given in a form that is easy to understand.

We will soon see, however, that this program and computers have limitations. Computers will only do what they are programmed to do. If you give the computer the wrong kind of information, you'll get strange results. There is an equation that applies here, "garbage in = garbage out," or GIGO for short.

2. * INPUT ERROR

This is an error message from the computer. The computer was expecting a numeral from you, and you typed ENTER without a numeral. The computer printed * INPUT ERROR, because it was waiting for a numeral. After the computer prints * INPUT ERROR, you can go ahead and type a number.

3. * INPUT ERROR

Again you typed something the computer was not expecting. The computer wanted numbers, and you typed characters. To the computer, FOUR does not equal 4.

The second program uses string variables with the INPUT statement. Remember that string variables can contain characters.

4. If your name is Melanie, the computer will reply
   PLEASED TO MEET YOU MELANIE

Many programs use an INPUT routine like this to appear friendly. It also makes the computer appear to be smart. As you'll see, however, the computer is not intelligent.

5. PLEASED TO MEET YOU 1234

6. The computer does not know if you used a false name.

Notice that the program did not "bomb" when you typed numerals instead of letters. This is because string variables can contain any characters, including numbers.

7. The REM statements don't show up in the program.

The purpose of the REM statement is to make the program easier to understand when someone else reads your program or when you come back to it later.

The input sections of programs are very important. As you can see, the computer cannot handle certain types of information. Other types of information cause the computer to give wrong or misleading information.

A program can be designed to minimize input errors. Some of the techniques are discussed in the software section of this book.

LESSON 6

In the lessons so far, you have run programs that have executed the instructions in the order that they are written in the program. Sometimes you'll want certain instructions to be followed over and over, or you may want to jump to a group of instructions if certain conditions are met.

One of the statements that will cause instructions to be followed out of order is the GOTO statement. Using GOTO, however, can cause you problems. One problem is that you can get into an endless loop, and your program goes around in circles.

The first program on the worksheet contains an endless loop. You will know when it happens. The computer screen will fill with information and the cursor will disappear. You cannot type other instructions into the computer until you see the cursor on the screen again.

To stop the program and get the cursor back, you'll type FCTN-4. Push down the FCTN key while you push the "4" key. If you do it right, the program will stop, and you'll again see the READY message.

□ Discussion

The first program is fun to use in introducing people to the computer. It's fast paced with lots of action!

1. If your name is Dick, the screen will rapidly fill with

WORKSHEET 6

TYPE THIS ON THE COMPUTER THEN	ANSWER THESE QUESTIONS
NEW 1Ø CALL CLEAR 2Ø INPUT "WHAT IS YOUR NAME ": NAM$ 3Ø PRINT "HI "; NAM$; 4Ø GOTO 3Ø	1. What happened? 2. Why?
FCTN/4	3. What did FCTN/4 do? __________
3Ø PRINT NAM$; " is "; 4Ø GOTO 8Ø 5Ø PRINT "INTELLIGENT" 6Ø PRINT "FRIENDLY" 7Ø PRINT "POPULAR" 8Ø PRINT "A COMPUTER GENIUS"	Notice the blank before and after the word IS. 4. What message was printed? _____ ______________________________ 5. Were the instructions in lines 5Ø, 6Ø, and 7Ø executed? ______________
4Ø LIST RUN	Type 4Ø and press RETURN 6. Is line 4Ø still there? ____________ 7. What was printed? ____________ ______________________________
4Ø GOTO 6Ø	Experiment with different GOTO values to see what messages the computer will print.

"HI DICKHI DICKHI DICK" and so on.

2. The GOTO statement causes line 30 to be executed over and over in an endless loop.

Line 10 clears the screen. Line 20 asks for your name. Line 30 says HI to you. Notice the semicolon at the end of the line. It causes the printing to continue on the same line until the line fills up. Line 40 sends control back to line 30, which prints another greeting. Lines 30 and 40 make the endless loop that prints greeting after greeting.

3. FCTN-4 stops the program.

Computer beginners often have trouble typing FCTN-4. Remember to keep the FCTN key pressed while you push the "4" key. If the program does not stop, you'll have to try again.

The second program uses lines 10 and 20 from the first program. Remember that typing new lines 30 and 40 erases the old lines 30 and 40.

4. If your name is Bev, the message is, "BEV IS A COMPUTER GENIUS."

5. Lines 50, 60, and 70 were not executed.

The GOTO statement caused the computer to skip over lines 50, 60, and 70, and go to line 80.

6. Line 40 is gone.

The purpose of typing 40 was to erase the old line 40. The new line 40, which contained nothing, replaced the old line 40, which was 40 GOTO 80.

7. CHRISTY IS INTELLIGENT
FRIENDLY
POPULAR
A COMPUTER GENIUS

With the GOTO 80 statement removed from the program, the computer followed the instructions in order, printing out all the messages.

Now try different GOTO statements to see what happens. If you get caught in an endless loop, remember FCTN-4.

LESSON 7

The computer works well for repetitive tasks like counting. Several kinds of tasks involve counting. In a drill program, the computer might count the number of problems to present to a student. Each time a correct answer is scored, 1 could be added to the score. Finally the computer might count the length of time that a message appears on the screen.

As you do the worksheet, at least one thing is going to look strange to you:

```
N = N + 1
```

If N = 5, then you have the equation,

```
5 = 6
```

That doesn't look right, does it? The computer, though, has no problem understanding. When it sees N = N + 1, it finds N's memory location and adds 1. N used to equal 5. Now it equals 6.

Another way to think of the equals sign is to think of it meaning "is replaced by." N = N + 1 would then read N is replaced by N + 1.

□ Discussion

The first example shows that counting for humans is a slow process.

1. 1,2,3,4,5,6,7,8,9,10

There has got to be an easier way to count! The second program uses N = N + 1 to count.

WORKSHEET 7

TYPE THIS ON THE COMPUTER	THEN ANSWER THESE QUESTIONS
NEW 10 CALL CLEAR 20 REM COUNT TO 10 30 PRINT " 1,2,3,4,5,6,7,8,9,10 "	1. What was printed? ____________ ____________________
30 N = 1 40 PRINT N 50 N = N + 1 60 IF N = 11 THEN 80 70 GOTO 40 80 END	2. What was printed? ____________ ____________________
30 FOR N = 1 TO 10 40 PRINT N 50 NEXT N 60 70 80 RUN	You will be replacing part of the above programs and deleting some lines. 3. What was printed? ____________ ____________________
NEW CALL CLEAR 20 REM COUNT TO 3000 30 PRINT "COUNTING TO 3000" 40 FOR N = 1 TO 3000 50 NEXT N 60 PRINT "FINISHED"	Sometimes the computer counts silently. 4. About how long was the delay between "COUNTING TO 3000" and "FINISHED"? ____________ ____________________

2. 1 2 3 4 5 6 7 8 9 10 (printed vertically)

The program uses a loop. At first N = 1. Next N is printed. N = N + 1 makes N = 2. GOTO 40 causes N, which equals 2, to be printed. This process could continue for a long time, adding one to N and printing the new value. Line 60 was added to stop the program after 10 was printed. N = 11, but the program was stopped before the 11 could be printed.

Notice in this worksheet, as in some past worksheets, that you don't type in a completely new program each time. The second and third programs each use lines 10 and 20 from the first program.

3. 1 2 3 4 5 6 7 8 9 10 (again vertically)

This program uses what is called for a FOR. . .NEXT loop. The loop starts with the FOR statement and continues until the NEXT statement. The FOR part of the loop determines how many times the loop will be executed. In this program, we have 30 FOR N = 1 to 10. The starting value of N is 1, the ending value will be 10. Each time through the loop, N is increased by 1.

The first time through the loop, N = 1, which is printed. The second time through the loop, N = 2 and is printed. This continues until N = 10 and is printed.

The computer works so fast that it's easy to think that the computer carries out instructions instantaneously. This is not true. It takes a split second to carry out each instruction.

The last program uses a FOR. . .NEXT loop to show two things. First, it does take time for the computer to execute instructions. Second, here is a program you can use when you need a time delay. For instance, you have some text that you want to display for a definite time, before showing more text.

4. The delay was about 9 seconds.

In this program, the computer "counted" to 3000. There was no indication that the computer was counting, because the numerals were not printed on the screen. Nevertheless, the computer started out with N = 1. Then N = 2, and so on, up to N = 3000. To complete the task, the computer took a little over 9 seconds.

The FOR. . .NEXT loop is good to experiment with. It can count by numbers other than 1 with the STEP command. Try this one:

```
10 FOR N = 0 TO 20 STEP 2

20 PRINT N

30 NEXT N
```

If you like to count backwards, here is another program:

```
10 FOR N = 10 TO 0 STEP -1

20 PRINT N

30 NEXT N
```

LESSON 8

One of the most entertaining features of a computer is its graphic capabilities, which allow you to show pictures on the screen. For instance, if you are writing a program that teaches addition, you could use graphic demonstrations of the addition process.

Several commands will allow you to do many interesting things with your TI computer. We will use some of the simpler commands in this lesson to show you some of the nice effects you can create without getting into anything too complicated.

Before you start the worksheet, there are a few things to keep in mind. First, some of the graphics commands only work in the program mode. Once a program has run its course, the computer reverts back to its original conditions. The CALL SCREEN(N) command is one such command.

The next thing to notice is the GOTO statement at the end of each program. This keeps the program running so that you can see what happened. You'll remember that this is called an "endless" loop, and the way to get out of it is to press FCTN-4.

Finally, this exercise will be most enjoyable with a color television or monitor. If you are working with a black and white or green display, the commands will work, you just will not be able to see the different colors.

□ Discussion

1. The screen is red (if your television is adjusted properly).

The TI gives you 16 different colors to choose from. Those colors are numbered as follows:

1. TRANSPARENT
2. BLACK
3. MEDIUM GREEN

9. MEDIUM RED
10. LIGHT RED
11. DARK YELLOW

WORKSHEET 8

TYPE THIS ON THE COMPUTER THEN ANSWER THESE QUESTIONS

```
NEW
10 CALL CLEAR
20 CALL SCREEN(7)
30 CALL COLOR(2,16,1)
40 CALL HCHAR(12,1,42,32)
50 CALL VCHAR(1,16,42,24)
60 GOTO 60
```

REMEMBER: FCTN/4 stops an endless loop.

1. What color is the screen? ______
2. What happens if you change the 16 in line 30 to a 2? ______
3. How many stars are in the horizontal (back and forth) line? ______
4. If you change the 42 in line 50 to a 43, what happens to the vertical (up and down) line? ______

______

```
NEW
10 CALL CLEAR
20 CALL SCREEN(13)
30 CALL COLOR(2,16,1)
40 LINE = 1
50 FOR X = 16 TO 1 STEP −1
60 Y = 17 − X
70 CALL HCHAR (Y,X,42,LINE)
80 LINE = LINE + 2
90 NEXT X
100 GOTO 100
```

5. What would you change in line 20 to make the screen red? ______
6. What would happen if the 2 in line 80 were changed to a 1? ______
7. Which variable tells the computer how many stars to put in a line?

______

______

8. What changes if you make line 100 say: 100 GOTO 10

______

______

4. LIGHT GREEN
5. DARK BLUE
6. LIGHT BLUE
7. DARK RED
8. CYAN
12. LIGHT YELLOW
13. DARK GREEN
14. MAGENTA (PURPLE)
15. GRAY
16. WHITE

TRANSPARENT and usually not used with the CALL SCREEN(N) command.

2. The "stars" or asterisks are black instead of white.

The CALL COLOR command has three numbers enclosed in parentheses following it. The first number, a 2 in our program, designates a set of 8 characters. There are a total of 16 "sets." Set 2 consists of:

SET #2

40	(
41	)
42	*
43	+
44	,
45	–
46	.
47	/

The other sets (1 and 3 through 16) are listed in the *TI User's Reference Manual.* The numbers preceding the characters are called ASCII (pronounced ASK-KEE) numbers. The second number following the CALL COLOR command sets the foreground color for the whole set. The third number sets the background color. You can refer to the screen color chart for a list of the color numbers.

In our example we set the foreground, or character color to 16, which is white. The background color was the transparent color, 1. The transparent color does not show up. If we had used 2 as the background color, each star would have been surrounded by a black square. One thing to remember is that once you set these up using the CALL COLOR command, it changes *all* of the characters in that set until the program ends or you use the CALL COLOR command again for that set. This includes characters that are already on the screen. Let's say you had a message on the screen that said "(6+8)/(7+7)=1." If you then (in your program) used the comniand CALL COLOR (2, 7, 16) so you could put some red stars surrounded by white squares on the screen, the parentheses, plus signs, and division signs also would turn red with white squares. They are also in SET 2.

3. There are 32 asterisks in the horizontal row.
4. The asterisks change to plus signs if you change the 42 in line 50 to a 43.

The CALL HCHAR and CALL VCHAR commands are very useful for creating pictures on the screen where you want them. Four numbers follow each of these commands.

The first number following either of these commands tells the computer how many rows DOWN from the top to move. The second number tells the computer how many columns over from the right to move. The top row (rows go across and columns go up and down) is row

1. The bottom row is row 24. The column on the left is column 1 and the far right column is column 32.

The third number is the ASCII number of the character that you want printed. A 42 is our old friend, the asterisk.

The last number tells the computer *how many* characters to print. This number is optional. If it is left out, the computer will print one of the characters that you designated at the location given.

5. You would change the 13 to a 7 to make the screen a dark red, a 9 to make the screen a medium red, or a 10 to make the screen a light red.
6. If the 2 in line 80 were changed to 1, the number of asterisks in a line would only increase by 1 each time the computer passed through the loop. The picture then would look like half a Christmas tree.
7. The variable LINE tells the computer how many stars to put in a line.
8. If you change line 100 to say 100 GOTO 10, the tree will be drawn, erased, drawn again, erased again, drawn again, etc.

Try using some of these commands to make some picture of your own. It is usually easier if you draw the pictures first on a sheet of graph paper. A sheet has been supplied (see Appendix A).

LESSON 9

Often you will write a program that needs outside information to run. An example is the INPUT statement, "WHAT IS YOUR NAME?" You can write the program in a general fashion and then have the information supplied when the program is run.

Other times, you will write a program and have all the information you need for the program. However, you will want a program that can be run with different sets of information or data. Maybe you are writing a test-giving program that will be used with different sets of questions. In such cases, the DATA statement is used.

By this time, you should be familiar with error statements from the computer, such as "* CAN'T DO THAT." Still, a few reminders will not hurt. First, if you try to run a program and get an error message, you will have to make corrections. Usually, that will take care of the problem. If your program is a hopeless mess, you can type NEW and start over again.

Do not hesitate to ask others to help you when you do not understand. Sometimes, even a noncomputer person can help you by reading the directions over with you.

WORKSHEET 9

TYPE THIS ON THE COMPUTER THEN	ANSWER THESE QUESTIONS
NEW 10 DATA 1,2,3 20 READ A,B,C 30 PRINT A,B,C	1. What was printed? ______ 2. Where did the numbers come from? ______
10 CALL CLEAR 20 FOR N = 1 TO 5 30 READ A 40 PRINT A 50 NEXT N 60 DATA 4,5,6,7,8	3. What was printed? ______ 4. Does it matter if the DATA statement is at the beginning or the end of the program? ______
NEW 10 CALL CLEAR 20 REM TEST PROGRAM 30 DATA "WHAT COLOR IS RUDOLPH'S NOSE?", "RED" 40 READ Q$,A$ 50 PRINT Q$ 60 INPUT RES$ 70 IF RES$ = A$ THEN 100 80 PRINT "NO, THE ANSWER IS "; A$ 90 GOTO 110 100 PRINT "GOOD JOB!" 110 END	Try this program twice, getting the answer correct and getting the answer incorrect. 5. What was printed when you got the answer correct? ______ ______ 6. What was printed when you got the answer wrong? ______ 7. What did Q$ equal? ______ 8. What did A$ equal? ______ 9. What would you have to change in the program to ask a different question? ______ ______ ______ ______

□ Discussion

Notice in these examples that three statements go together. They are DATA, READ, and PRINT.

1. 1 2
3

2. The numbers came from the DATA statement.

To print different numerals, you can change the DATA statement.

Often, you can write a program with one set of data. Then, you can use the same program, but with different data.

3. 4 5 6 7 8 (printed vertically)
4. It does not matter if the DATA statement is at the beginning or end of the program.

Notice the use of the FOR. . .NEXT loop to print the values of "A." It is useful when you are repeating something a definite number of times.

You would think that the DATA statement would have to be at the beginning of the program. It does not matter, however. The data can be anywhere in the program. When you give the READ command, the computer searches for the DATA line.

The last program uses the DATA statement to give a test. This is not a complete test program, but just a sample to show you what can be done. Notice the quotation marks in the first DATA statement. The quotation marks are not necessary in the example given. They are needed if the DATA item contains a comma, colon, or leading spaces.

5. "GOOD JOB" was printed.
6. "NO, THE ANSWER IS RED" was printed.
7. Q$ = "WHAT COLOR IS RUDOLPH'S NOSE?"
8. A$ = "RED"
9. You would change the DATA statement.

This is the main point of the lesson. Using the DATA statement, you can write a program that can be easily changed. For instance if you want to ask the question, "Who is buried in Grant's tomb?" change line 30:

```
30 DATA "WHO IS BURIED IN GRANT'S TOMB?", GRANT
```

LESSON 10

One of the factors that makes a game like Monopoly fun is the uncertainty of what happens next. You roll the dice to obtain a random number, for example an "8." Then you move eight spaces.

The computer has statements that accomplish the same purpose. The RND statement returns a random number. To direct the computer to do something based on the random number returned, you use the ON GOTO command.

Both RND and ON GOTO have uses other than games. As in the other worksheets, the purpose of this one is to introduce you to various commands. As you learn to program, you will continue to learn new commands, as well as find new uses for the old commands.

WORKSHEET 10

TYPE THIS ON THE COMPUTER THEN ANSWER THESE QUESTIONS

```
NEW
1Ø CALL CLEAR
2Ø REM RANDOM NUMBERS
   DEMO
3Ø X = RND
4Ø PRINT X
```

1. What did the computer print? ____

____

```
25 FOR N = 1 TO 1Ø
5Ø NEXT N
```

2. How many numbers were printed?

____

3. What was the lowest, highest? ____

```
3Ø X = INT (RND*1Ø)
```

4. What number was lowest, highest?

____

____

```
NEW
5 RANDOMIZE
1Ø CALL CLEAR
2Ø REM RANDOM COMMENTS
3Ø X = INT (RND*3)+1
4Ø ON X GOTO 5Ø,7Ø,9Ø
5Ø PRINT "YOU ARE
   WONDERFUL!"
6Ø GOTO 1ØØ
7Ø PRINT "I LIKE YOU!"
8Ø GOTO 1ØØ
9Ø PRINT "YOU'RE DOING
   GREAT!"
1ØØ PRINT "":"":"":"":"":"":"":"":"":
    "":""
```

Run this program several times.

5. Were the same comments printed each time? ____

____

6. Is it possible for two comments in a row to be the same? ____

____

7. If X = 1, what comment is printed?

____

____

8. Where could you use random comments in a program? ____

____

9. How would random numbers be useful in a game program? ____

____

____

□ Discussion

The first program will print out a random number. If the program is run again, it will run another random number.

1. The computer will print out a decimal.

2. The computer printed 10 numbers.

3. The lowest and highest numbers will vary.

Random numbers generated by the computer will vary from 0 to just less than 1. They may be all different or some of them may be the same. For most applications, the decimal random numbers will not be of much help. Usually, whole numbers in a certain range are needed. The next program takes care of that.

4. The lowest and highest number will vary.

All the random numbers are whole numbers in the range of 0 to 9. This set of numbers would be helpful when drilling on basic math facts.

Notice in line 30 that a couple of things have changed from the first RND statement. First, there is the "∗10." The RND is multiplied by 10. This gives a range of mixed numerals from 0 to less than 10. The other difference is the INT statement, which stands for "integer." It gives a whole number value less than or equal to the mixed numeral. For example, the INT value of 9.987056 is 9.

If you want random numbers in the range of 1 to 10, then the statement X = INT (RND∗10)+1 will be useful. If you want random numbers from 1 to 3, then change the 10 to a 3 in the above statement.

The last program uses the RND function and the ON GOTO statement. The value of the random number is used to cause different things to happen.

5. Different comments could have been printed.

6. It is possible for two comments in a row to be the same.

7. "YOU ARE WONDERFUL" is printed.

Let's look at this program more closely. The statement X = INT (RND∗3) + 1 generates random numbers in the range of 1 to 3. The random numbers are used by line 40, ON X GOTO 50, 70, 90. What this means is that if X = 1, then GOTO line 50 and print "YOU ARE WONDERFUL." If X = 2, then GOTO line 70 and print "I LIKE YOU," and so on.

8. They could be used in a drill program when a student answers correctly.

Anytime in a program where you want to give positive reinforcement, giving random comments will add variety to the program. Random comments can be combined with the student's name to personalize the message. A comment like "WAY TO GO, KEN" can really be encouraging.

9. Random numbers would be useful whenever you want something unpredictable.

In the simulation program LEMONADE, unpredictable events such as heat waves, street closings, and thunderstorms add excitement to the game. Drill-type games use random numbers to present questions in a random order.

LESSON 11

Sometimes when you are writing a program, a large number of variables may be related. You may want to refer to these variables in order without having to name each one. If you are reading questions and answers from DATA statements and using a different variable name for each question and each answer, it can be time consuming. You will have to program the reading of each question and answer separately. For instance, the first question could be A$ and the first answer might be B$. The second question and answer could be C$ and D$, and so on.

Wouldn't it be nice to write a program that would read the questions and answers in a loop and assign variable names to each question and answer. It would sometimes be useful to not have to specify the number of questions, and let the computer figure that out. When you write out a test, you have question 1, question 2, and so on. Is it possible to have a variable 1, variable 2, and so on?

Yes! Using arrays, it is possible to specify a variable name and then use that variable name with the numerals 0, 1, 2, 3,. . . to specify as many unique variables as you need. If you are reading questions, the first question might be QUES$(1), the second question, QUES$(2), and so on.

Before you use an array, you will need to dimension the array with the DIM statement. This will tell the computer how many elements your array has and will reserve space in the computer memory. If you are reading questions from DATA statements, DIM will specify how many questions there are. You will see the DIM statement in the worksheet. It does not have to be exact, as long as it is larger than it needs to be.

Arrays may be difficult for you to work with at first. It probably will take a while for you to appreciate the power they give you in writing programs. Just keep at it!

□ Discussion

In the first program, notice the DIM statement. It tells us that the variable A is being dimensioned:

WORKSHEET 11

TYPE THIS ON THE COMPUTER THEN	ANSWER THESE QUESTIONS
NEW 10 DIM A(5) 22 A(2) = 6 24 A(3) = 7 20 A(1) = 5 : 30 PRINT A(1),A(2),A(3)	1. What did the computer print? ____________ 2. Could you have used A,B,C for variable names? ____________
30 FOR N = 1 TO 3 40 PRINT A(N) 50 NEXT N LIST	3. What variable is printed first? ____ 4. How many numbers are printed? ____________
(ALPHA LOCK DOWN) NEW 10 DIM A$ (10) 20 CALL CLEAR 30 REM READ THE DATA 40 N = 1 50 READ A$(N) 60 IF A$(N) = "NO MORE" THEN 90 70 N = N + 1 80 GOTO 50 90 REM PRINT THE DATA 100 FOR J = 1 TO N − 1 110 PRINT A$(J) 120 NEXT J 130 DATA I, AM, LEARNING 140 DATA TO, PROGRAM 150 DATA NO MORE	5. Is A$(N) a numeric or string variable? ____________ 6. Will this program work with different DATA words? ____________ 7. Can you use more or fewer DATA words? ____________ 8. How does the computer know when to stop reading DATA? ____________ 9. What does N equal in line 100? ____________ 10. What does N represent? ________ 11. If N = 1 and J = 1, do A$(N) and A$(J) represent different variables? ____________

```
10 DIM A(5)
```

The number in parentheses is called the subscript. In this case the subscript is a 5. This means that the largest subscript in an array will be a 5. In the first program are 3 array variables, A(1), A(2), and A(3). You are allowed more variables, A(4) and A(5), as well as A(0), but it is fine to use fewer array variables than the DIM statement allows. Now for the answer to the first question:

1.
```
5     6
7
```

2. Yes, you could have used A, B, and C for variable names.

The first program does not show the use of arrays. It was meant to show you what they look like and how to use the DIM statement. The second program will show one of the advantages of arrays.

3. A(1) is printed first.
4. There were 3 numbers printed.

In the first program, each variable was mentioned by name: A(1), A(2), and A(3). In the second program, the variable name is A(N). Each time through the FOR. . .NEXT loop, A(N) names a different variable. The first time through the loop, N = 1 and A(N) is A(1). The second time through the loop, N = 2 and A(N) is A(2). If you wanted to print A(0) through A(100), it would be even nicer not to have to call every variable by name.

The last program is meant to show how you can read DATA items into an array and then have them printed out.

5. A$(N) is a string variable.

You can recognize A$(N) as a string variable because of the dollar sign. Notice in the DIM statement that A$ is dimensioned as 10. If the first array variable is A$(1), then you will be able to read 10 DATA items until you have A$(10).

Actually, the computer dimensions an array for you as 10. If your array has fewer than 10 array elements, you do not need a DIM statement. The reason the DIM statements was used on the worksheet was so that you could become familiar with it.

6. Yes! That is the point of the lesson.
7. Yes! If you use more than 10 DATA items, you will have to change the DIM statement.

Hopefully, you will see that using arrays along with the DATA statement and the FOR. . .NEXT loop give you a great deal of flexibility in using one program format with different information.

8. As soon as "NO MORE" is encountered, it is time to go to the print section of the program.

It is common in programs to read DATA in a loop. The computer continues to read data until a predetermined codeword or numeral is reached.

9. N = 7

10. N represents the number of DATA items.

You may be wondering about the "N – 1" in line 100. The seventh DATA item is "NO MORE." Since the message to be printed is "I AM LEARNING TO PROGRAM," we will want to go through the print loop only six times, which is "N – 1."

11. They will represent the same variable.

LESSON 12

When you are designing a program, it is helpful to plan the general program ideas first. In this worksheet, a simple addition drill program is presented. Here is is a simple program outline:

```
PRINT PROBLEM

INPUT

RESPOND
```

The first two sections of the program are straightforward and easy to program. The RESPOND section is more difficult, because a decision is involved. After the student answers a problem, the computer decides if the answer is correct. If the answer is correct, a positive response is made. If incorrect, a negative response is made. The outline of the RESPOND section will be:

```
IF ANSWER IS INCORRECT
    THEN POSITIVE RESPONSE
    ELSE NEGATIVE RESPONSE
```

As you subdivide your programs, it is useful to program some of the subdivisions as subroutines. A subroutine is a unit of your program. It is called with the GOSUB command. At the end of the subroutine, the RETURN command returns control to the main program.

□ Discussion

This program may be too long for you to type in one session. If so, type as much of the program as you want and then save the program either on disk or tape. If you are using a disk drive, first think of a file name. How about "DRILL +"? Then type:

```
SAVE DSK1. DRILL +
```

If you are using a tape recorder, type:

```
SAVE CS1
```

When you are ready to continue typing, load the program. If using disk, type:

```
OLD DSK1. DRILL +
```

if the name of your program is "DRILL+." Then LIST the program and start typing where you left off.

If you are using a tape recorder, type:

```
OLD CS1
```

1. The REM statement causes no action.

Notice that the REM statements show the organization of the program in three sections: PRINT PROBLEM, INPUT, and RESPONSE. The last three REM statements show the three subroutines: POSITIVE RESPONSE, NEGATIVE RESPONSE, and PRESS ENTER.

2. A can take the value of 0 through 9.
3. The second addend will be a 9.
4. C represents the sum of A and B.

This program will drill on "9s" addition. Other programming options are available. Changing line 40 to:

```
40 B = INT (RND*10)
```

would drill on all the addition facts through "9."

WORKSHEET 12

TYPE THIS ON THE COMPUTER THEN ANSWER THESE QUESTIONS

```
NEW
5 RANDOMIZE
1Ø REM PRINT PROBLEM
2Ø CALL CLEAR
3Ø A = INT(RND*1Ø)
4Ø B = 9
5Ø C = A + B
6Ø PRINT A;“+”;B;“=”;
7Ø REM INPUT
8Ø INPUT TRY
9Ø IF TRY <> C THEN 14Ø
1ØØ REM RESPOND
11Ø GOSUB 2ØØ
12Ø GOSUB 4ØØ
13Ø GOTO 2Ø
14Ø GOSUB 3ØØ
15Ø GOSUB 4ØØ
16Ø GOTO 2Ø
2ØØ REM POSITIVE RESPONSE
21Ø CALL CLEAR
22Ø CH = INT (RND*3) + 1
23Ø ON CH GOTO 24Ø,25Ø,26Ø
24Ø PRINT “WOW”
245 GOTO 27Ø
25Ø PRINT “SUPER”
255 GOTO 27Ø
26Ø PRINT “NICE“
265 GOTO 27Ø
27Ø RETURN
3ØØ REM NEGATIVE RESPONSE
31Ø CALL CLEAR
32Ø PRINT “THE ANSWER IS ”; C
33Ø RETURN
4ØØ REM PRESS ENTER
41Ø PRINT “”:“”:“”:“”:“”:“”:“”:“”:“”:
    “”:“”:“”
42Ø PRINT “PRESS ENTER TO
    GO ON”
43Ø CALL KEY(Ø,K,S)
44Ø IF K=−1 THEN 43Ø
45Ø RETURN
```

1. What action does line 1Ø cause?
   ____
   ____
2. What values can A take? ________
   ____
3. What value will the second addend be? ______________________
4. What does C represent? ________
5. Why is there a semicolon at the end of line 6Ø? ______________
6. What is the purpose of line 22Ø?
   ____
7. In line 42Ø could you push a key other than RETURN to go on? ___
   ____
8. What is the difference between the GOSUB and GOTO commands?
   ____

5. The semicolon causes the "?" and cursor of the INPUT statement to appear on the same line as the problem.
6. Line 220 will generate a random number in the range of 1 to 3. If CH = 1 then "WOW" will be printed, and so on.
7. You could press almost any key to go on.

Some keys, such as SHIFT by itself, would cause no action. The CALL KEY command is used here. CALL KEY works the same as INPUT, with two exceptions. First, you do not have to press ENTER. Second, you can "get" only one character at a time, instead of a whole line as with INPUT. Sometimes you will see programs that ask you if you want to run the program again. You type Y or N, and before you press ENTER, something happens. Usually the CALL KEY command is used in this case.

The CALL KEY statement needs three things in the parentheses following it. First of all is the 0 followed by a comma. The next thing is two variables separated by a comma (K and S in our program). After executing a CALL KEY (0, K, S) command, the variable K will contain a −1 if no keys were pressed, or the ASCII code of a key that *was* pressed. The S will contain a 0 if no key is being pressed, a 1 if the key pressed was a new one, or a −1 if the key pressed is the same as before.

8. GOTO sends control to another part of the program. GOSUB sends control to a subroutine and then back.

It is like the difference between a one-way ticket and a round-trip ticket. The round-trip ticket, like GOSUB, allows going elsewhere, doing something, and returning.

chapter 6
PROGRAMMING IN LOGO

LOGO LESSON 1

Before you start the first lesson, we need to discuss a few things. First, familiarize yourself with the pages in the TI-LOGO manual that show you how to set up and start LOGO on your TI.

Next, take a look at the first worksheet. The worksheet is divided in half. Type the part on the left side of the sheet. One of the lines says:

```
FD 80 !E!
```

On the computer, type:

```
FD 80
```

and then press the ENTER key, as shown by the !E! on the worksheet. Be careful to type the lines the way they appear on the worksheet. Notice the space between the FD and 80.

If you make a mistake in typing, use FCTN-3 to erase. If you discover the mistake after you press ENTER, retype the line.

In the first worksheet, you'll start exploring turtle graphics. The triangle you'll see on the screen represents a turtle. As you move the turtle around the screen with various instructions, the turtle leaves tracks and draws pictures. When you want to draw a different picture or start over to fix a mistake just type CS and press ENTER.

If you get stuck on the worksheet, look ahead to the discussion section. It won't be cheating to look at the answers. Sometimes you may need to go over a worksheet several times before you understand it.

Now load TI LOGO. When you see:

```
WELCOME TO TI LOGO!
```

LOGO WORKSHEET 1

INSTRUCTIONS	QUESTIONS
TELL TURTLE !E!	1. What happened? ________
FD 8Ø !E! BK 4Ø !E! RT 9Ø !E! FD 3Ø !E! LT 9Ø !E! FD 4Ø !E! BK 8Ø !E! RT 9Ø !E! PU FD 3Ø !E! LT 9Ø !E! PD !E! FD 8Ø !E!	2. What does FD do? ________ 3. Does RT move or rotate the turtle? ________ 4. What action does BK cause? ________ ________ 5. What does PU do? ________ 6. What does PD do? ________ 7. What message was printed? ________
CS !E!	8. What happened? ________
FD 8Ø !E! RT 9Ø !E! FD 1Ø !E! RT 9Ø !E! FD 8Ø !E! RT 9Ø !E! FD 1Ø !E!	9. What is drawn? ________
CS !E!	

you're ready to begin. The first instruction on the worksheet, TELL TURTLE, should display the triangle (turtle) in the middle of the screen. Don't forget, !E! means to press the ENTER key.

□ Discussion

Turtle graphics is fun, isn't it? You can draw pictures on the screen without knowing a lot about the computer.

1. The triangle (turtle) appeared on the screen.
2. FD moves the turtle forward.

The forward direction is figured in relation to the turtle. For us, it may look like the turtle is moving to the left or the right.

3. RT rotates the turtle.

At first the RT and LT commands can be confusing. It's natural to want to move the turtle to the right, instead of merely rotating it. To move to the right, you have to give the RT 90 command, followed by an FD command.

4. BK moves the turtle back.

Notice that the BK command doesn't erase the turtle tracks already drawn.

5. PU stops the turtle from drawing.

PU stands for pen up. When you draw on paper with a pen, you sometimes have to lift your pen to start drawing in a different place. It's the same with the turtle. You use PU so you can move the turtle to a different part of the screen to start a new part of your drawing.

6. PD starts the turtle drawing again.

After you've used PU to move the turtle, you have to type PD (pen down) to resume drawing.

7. The message "HI" was printed.

The turtle can be used, as in this case, to print messages, as well as pictures.

8. CS clears the screen.

Besides clearing the screen between pictures, CS clears out mistakes.

9. A rectangle is drawn.

Turtle graphics is particularly well-suited for drawing geometric figures. The turtle, besides being playful, can help you visualize many geometric concepts.

LOGO LESSON 2

In this lesson, you'll explore more geometric figures and learn a short cut that will save you typing time.

In the last lesson, the !E! symbol appeared at the end of each line to remind you to press the ENTER key. The !E! symbol won't be used anymore. Just remember to press ENTER at the end of each line.

LOGO WORKSHEET 2

INSTRUCTIONS	QUESTIONS
TELL TURTLE FD 50 RT 120 FD 50 RT 120 FD 50 RT 120	1. Does each instruction have to start on a new line?__________ 2. What is drawn?______________
CS	
REPEAT 3 [FD 50 RT 120]	3. What is drawn?______________
CS	
REPEAT 360 [FD 1 RT 1]	4. What is drawn?______________ 5. Is it shaped perfectly? ________ 6. Is it drawn slowly?__________
REPEAT 20 [FD 18 RT 18]	7. Does this figure look almost like the last one? ______________ 8. Is it drawn faster? __________
CS	
RT 90 REPEAT 10 [FD 10 LT 18] REPEAT 10 [FD 10 RT 18]	9. What is drawn?______________ 10. What added instructions would make an "8"?______________
CS	

As you do this lesson, don't forget to type a space between the letters and numbers of such commands as RT 120. You'll be seeing two new symbols starting with this lesson. They are the "[" and "]" symbols. To type them, hold down the FCTN key and type an "R" for "[" and "T" for "]." If you make a mistake in your picture, remember that typing CS will give you a new chance.

□ Discussion

The advantage of turtle graphics over other types of graphics is that you don't have to understand coordinates. You just tell the turtle which way to point and how far to travel.

1. Instructions don't have to start on a new line.

Most of the instructions on these worksheets start on a new line, but don't have to. The important thing to remember is to leave a space between separate instructions.

2. A triangle is drawn.

Two things you might have noticed. First, the triangle isn't sitting on its base. This is because the turtle starts drawing in a vertical position. If you want the triangle to sit on its base, type LT 90 and then follow the instructions on the worksheet to draw a triangle.

The other thing you might have noticed is that an equilateral triangle, a triangle that has all sides equal, has three 60-degree angles, and you typed RT 120 to draw the triangle. The angle inside the triangle is called the interior angle. It measures 60 degrees. The angle that the outside of the triangle makes is called the exterior angle. For an equilateral triangle, the exterior angle is 120 degrees. This is also the angle that the turtle has to turn as it draws the triangle.

3. Again, a triangle is drawn.

The REPEAT command is a time-saving feature of LOGO. Usually when you draw geometric figures, you repeat instructions. To use REPEAT, specify the number of times you want an action repeated. Then enclose the action in brackets "[]".

4. A circle is drawn.
5. It is not shaped perfectly.
6. It is drawn slowly.

Here is a perfect example of the usefulness of the REPEAT command. It would be very time consuming to type 360 instructions of FD 1 RT 1.

The circle isn't drawn perfectly because of the limitations of the computer. Instead of drawing a continuous curve, the computer draws a series of dots or short lines. In this instance, the turtle actually draws a 360-sided polygon.

The circle is drawn slowly because the turtle has to stop 360 times to turn one degree. The LOGO commands operate slowly anyway, because the computer has to do a lot of processing for each command.

7. Yes, this figure looks almost like the last one.
8. Yes, it's drawn faster.

This circle is less of an approximation of a real circle. It's actually a 20-sided polygon. For many applications, however, it's close enough to a circle. The advantage of using it is that it draws faster, because the turtle has fewer turns to make.

9. The letter "S" is drawn.
10. The instructions are REPEAT 10 [FD 10 RT 18]
REPEAT 10 [FD 10 LT 18]

Go ahead and experiment with different polygons and circles. One thing that might help you is to remember that for the turtle to return to its starting point, its rotations must total a multiple of 360 degrees.

LOGO LESSON 3

One of the exciting features of the TI-LOGO is its use of sprites. A sprite starts out as an invisible object. As you assign the characteristics of shape, color, direction, and speed, the sprite becomes visible. Once the sprite becomes visible and starts to move, it will move independently of the turtle.

As you type the worksheet, notice the ":" in front of the word "truck" in the third line of the worksheet. The symbol is called "dots." The word "truck" has a value associated with it. Line three of the worksheet then means carry the value represented by the word "truck." When using dots, remember to type a space before, but no space after. Also notice that dots aren't used with numbers.

□ Discussion

Did you have trouble getting rid of all those sprites once you got them going? It can be frustrating to type CS and notice that the sprites are still going their merry way.

LOGO WORKSHEET 3

INSTRUCTIONS	QUESTIONS
TELL TURTLE TELL SPRITE 1 CARRY :TRUCK SETCOLOR :BLACK HOME	1. What do you see? ____________ 2. Where is it? ____________
SETHEADING :EAST SETSPEED 1Ø	3. Is the truck moving? ____________ 4. Which direction? ____________
TELL SPRITE 2 CARRY :ROCKET SETCOLOR :GREEN SETSPEED 1Ø	5. What do you see? ____________ 6. Which direction is it moving? ____ 7. Was a heading specified? ______
TELL SPRITE 3 CARRY :BALL SETCOLOR :RED HOME	8. Does the rocket go in front of or behind the ball? ____________
SETHEADING 45 SETSPEED 3Ø	9. Which direction is it moving? ____
CS	10. Did the sprites disappear? _____
TELL :ALL CARRY Ø SETCOLOR Ø SETSPEED Ø SETHEADING Ø	11. Now are the sprites gone? _____
TELL TURTLE	

1. A black truck is motionless.
2. The truck is in the middle of the screen.

TI-LOGO supports a maximum of 32 sprites. We're instructing sprite 1 take the shape of a black truck. The HOME position for the sprite, as well as the turtle, is the middle of the screen.

3. The truck is moving slowly.
4. The truck is moving from left to right.

The SETSPEED command can give the sprite a speed in the range of −127 to 127. Negative numbers move the sprite in reverse.

SETHEADING gives the sprite direction. The heading can be given in degrees or in the directions NORTH, SOUTH, EAST, and WEST.

5. I see a green rocket.
6. The rocket is moving upward.
7. No direction was specified.

Now two sprites are moving at the same time. Since no heading was specified for sprite 2, the heading is 0 degrees, which is NORTH.

8. The rocket goes in front of the ball.

This is an interesting feature of sprites. When they cross paths, the sprite with the lowest number covers the other sprite for a moment and appears to go in front. It's like a cloud moving in front of the sun and covering it up.

9. The sprite is moving from the lower left of the screen to the upper right.
10. CS did not clear the screen.

After a while, you get tired of seeing sprites all over the screen and want to get rid of them. CS, however, clears only the turtle tracks and text.

11. Yes, the sprites are all gone!

To clear the sprites, you have to first get their undivided attention. You do this with the TELL :ALL command. Then you tell them to carry nothing with the color CLEAR and with a speed ZERO.

The last command is TELL TURTLE. Now the ordinary turtle commands will work.

LOGO LESSON 4

The commands built into the LOGO language are called primitives. Examples are RIGHT and FORWARD. The primitives are useful for moving the turtle, but wouldn't it be nice if there were commands that would do things like draw geometric figures?

In LOGO, we can create programs called procedures that act like primitives. For instance, we could write a procedure called box. Then to make the box on the screen, we would type BOX.

The first procedure on the worksheet is called TRI. To write the TRI procedure, you'll use the LOGO editor. You'll type TO TRI and press ENTER twice. The first time you press ENTER, you'll enter the editor, and the screen will change color. The second ENTER will move the cursor down one line so that you can start typing the procedure. You'll notice that the procedure already includes an END statement. Make sure that you don't erase it.

If you need to correct mistakes, the most helpful keys are the arrow keys and the ERASE (FCTN 3) keys. To use the arrow keys, press the FCTN key and then the proper arrow key.

After you've typed the last line in the procedure key, don't press ENTER (if you do, it won't hurt, just press ERASE). Press BACK (FCTN 9) to leave the editor.

□ Discussion

Once you understand procedures, you'll do more drawing with less work. The procedure will allow you to draw complicated figures with just a few keystrokes.

1. The screen and the cursor changed color.
2. The turtle didn't draw.

When you type TO TRI, you entered the LOGO editor. You can tell by different screen color. While you're in the editor, the turtle is inactive.

3. A triangle was drawn.

The procedure TRI is now a part of the LOGO language in your computer, until you turn off the computer. If you want to save your procedures to tape or disk, check your TI-LOGO manual. You can draw designs combining TRI with the movement and rotation commands.

4. A design resembling a bow tie was drawn.

The LEAVES procedure used the TRI procedure just defined to make a more complicated design. This is part of the fun of LOGO: using procedures as building blocks in other procedures.

5. A hexagon-shaped figure was drawn.

Again the TRI procedure was used to draw a more complicated design.

Sometimes a builder will use a basic shape in building a building. The geodesic dome is a pattern of repeating triangles.

6. I see a flower.

LOGO WORKSHEET 4

INSTRUCTIONS	QUESTIONS
TELL TURTLE TO TRI (PRESS ENTER TWICE) REPEAT 3[FD 30 RT 120] (PRESS FCTN 9)	1. What happened?________________ 2. Did the turtle draw?______________
CS TRI	3. What was drawn? ________________
TO LEAVES (PRESS ENTER TWICE) RT 60 TRI RT 180 TRI RT 120 (PRESS FCTN 9)	
CS LEAVES	4. What was drawn? ________________
TO PETALS (PRESS ENTER TWICE) REPEAT 6[TRI RT 60] (PRESS FCTN 9)	
CS PETALS	5. What was drawn? ________________
TO FLOWER (PRESS ENTER TWICE) BK 45 FD 15 LEAVES FD 60 PETALS (PRESS FCTN 9)	
CS FLOWER	6. What do you see?________________

Think of how you draw a flower on paper. First you draw the stem. Then you draw the leaves and the petals. The FLOWER procedure is similar. The BK and FD commands draw the stem. Then the LEAVES and PETALS procedures are used to draw leaves and petals.

Extending the idea of a procedure within a procedure, we could design a procedure called GARDEN that would draw several flowers. Try it.

LOGO LESSON 5

In the last lesson, you used procedures to draw designs using triangles. These procedures, however, had a drawback. All the triangles were one size. In this lesson, you'll see how to draw figures that can change in size.

When you use the FD command, you give the number of turtle steps to move forward. With the CIRCLE :S procedure, you'll be able to draw different sized circles.

Another topic you'll explore is called recursion. One type of recursion is used in the procedure SLINKY. The last line of the procedure SLINKY is SLINKY. This will give interesting results.

When you are in the middle of a design, if you want to stop the turtle, press BACK (FCTN 9). If you have trouble with this lesson, review the use of the editor in lesson 4.

□ Discussion

If you have made it this far with the LOGO lessons, you should start to appreciate the power of the LOGO language. LOGO is a good beginning programming language, but it doesn't stop there. No matter how sophisticated a programmer you become, there are still ideas to explore using LOGO.

1. A small circle is drawn.
2. A larger circle is drawn.
3. The circle is so large that it won't fit on the screen.

This procedure uses the variable S, the length of one side of the circle. Actually, the circle is a decahedron, a 10-sided polygon. When you use the procedure CIRCLE :S, you type CIRCLE and then a number. This is similar to using the command FD 50. You type FD and then a number.

One of the problems with LOGO designs is that they sometimes don't fit on the screen. CIRCLE 100 is an example. If a part of the design disappears off the top of the screen, it will reappear on the bottom of the screen. This is called wrapping.

LOGO WORKSHEET 5

INSTRUCTIONS	QUESTIONS
TELL TURTLE TO CIRCLE :S REPEAT 1Ø [FD :S RT 36] (FCTN 9)	
CS CIRCLE 5 CIRCLE 35 CIRCLE 1ØØ	1. What is drawn? ________ 2. How is this different? ________ 3. What is wrong? ________
TO SLINKY CIRCLE 5 FD 5 LT 5 SLINKY (FCTN 9)	
CS SLINKY FCTN 9 (AFTER DESIGN IS COMPLETE)	4. What could be improved? ________ 5. What happened? ________
TO MOVE CS SX 5Ø SY 22 (FCTN 9)	
MOVE SLINKY FCTN 9 (AFTER DESIGN IS COMPLETE)	6. What happened? ________ 7. Did it stay on the screen? ________
TO DOUGHNUT MOVE SLINKY (FCTN 9)	
DOUGHNUT	8. What do you see? ________

4. The design could be placed in the middle of the screen.

The SLINKY didn't look right because it "wrapped" to different parts of the screen.

5. FCTN 9 caused the turtle to stop.

6. The procedure MOVE moved the turtle.

7. Yes, the design stayed on the screen.

Notice that CS was made part of a procedure. Also notice SX 50 and SY 22. The HOME position of the turtle has coordinates 0,0. SX 50 sets the X coordinate of the turtle to 50. This moves the turtle 50 steps to the right. SY 22 sets the Y coordinate of the turtle to 22 and moves the turtle 22 steps up. The turtle was placed on the screen in such a way that SLINKY would stay on the screen without wrapping.

8. Again, SLINKY is drawn in the middle of the screen.

The DOUGHNUT procedure combined MOVE and SLINKY into one procedure.

I hope you have learned enough about LOGO to want to explore it more. These worksheets have explored only turtle graphics. There is more to LOGO.

chapter 7
WRITING SOFTWARE

If you've worked your way through the 12 programming worksheets, then you're on your way to becoming a programmer. To further develop your skills, you'll have to study books, magazines, and other programs. Then you'll have to practice. It's a case of "use it or lose it." You may be wondering what programs to write.

You probably have seen that there are problems finding, evaluating, and using commercial software. Wouldn't it be nice if you didn't have to depend on others for your software needs? Are you interested in saving the money that software costs? Writing your own software has several advantages.

First, you'll be learning an important skill. Learning any skill sharpens your brain and helps you to think more clearly. Designing software should force you to be organized. The computer is not understanding or forgiving. It'll force you to give it instructions that it understands. "Oh, you know what I mean," doesn't work with a computer.

Writing software will give you some insights into the educational process. As you program a computer, you are, in a sense, teaching the computer. Seeing how the computer processes information gives you hints as to how humans think. For instance, in a tutorial, the computer will pose a question and wait for an answer. If the question is answered correctly, the computer probably will give an encouraging reply and move on. If the question is answered incorrectly, the computer will give a negative response and possible either give another chance at the right answer or review the material.

Let's compare that to a classroom scene. The class has been discussing simplifying fractions. The teacher is now asking class members questions.

> *"Christy, can you simplify 2/4?"*
>
> *"The answer is 1/2," replies Christy.*
>
> *"That's right! Good listening, Christy. Josh, simplify 6/8, please."*
>
> *Josh hesitates, "Is it 3/8?"*
>
> *The teacher feels that Josh is on the right track but needs a hint to get the correct answer.*
>
> *"You have the numerator correct. Divide both the numerator and denominator by 2 and you'll get the right answer," explains the teacher.*
>
> *"I get it," replies Josh. "The answer is 3/4."*

In the classroom, teachers have a strategy. They pose questions and give certain answers for correct questions and other replies for incorrect answers. Of course, teachers are capable of far more types of conversations than computers and have a better understanding of students. The point is that the computer program gives insights into human thinking.

As you learn to program the computer, you'll run into snags and probably become frustrated at times. Hopefully, your frustrations "teaching" the computer will help you be more understanding of young people. We adults sometimes forget the difficulties of learning. Young people are keenly aware that learning can be hard. Working with a computer reminds us of the difficulties of learning.

Another benefit of designing your own software is excitement. Writing a program is a real accomplishment. In some ways, it's like a game.

In a game, the outcome is not certain. You may win, and then again, you may lose. A game has hurdles to overcome. You have to avoid landing on the wrong square, striking out, or committing a foul. If you win the game, you feel excited and confident.

Writing software has its uncertainties, too. You can proceed in program development and suddenly run into a problem that causes you to totally abandon the project and try a different way.

As you write a program, you have to overcome hurdles. Some of the hurdles are messages from the computer, such as "CAN'T DO THAT," or "INCORRECT STATEMENT," or "FOR-NEXT ERROR." Other problems concern getting the computer to do something that to you, as a human, is obvious, but the computer just cannot seem to understand.

When you finally have a program that other people enjoy using that does what you designed it to, you've really accomplished something and have a right to feel proud.

If you enjoy writing software and learn to do it well, there are economic rewards. The quality of educational software is improving, but there is still a pressing need for high-quality software. As you study programming and education, you'll become more sensitive to the need for educational software and become aware of the capabilities of the computer for meeting those needs.

If you manage to sell a computer program, you'll either be paid a flat rate or receive royalties. If you're paid royalties, you may receive an income from the program for several years.

You don't have to be an accomplished programmer to receive another financial benefit. That is the benefit of saving money. Each program that you write and use saves you the money of buying the program. You have even more if you need multiple copies of a program. Since most commercial programs are copy protected, when you buy one program, one copy is all you get.

Besides being able to copy your own programs freely, you can change them to fit the needs of young people with whom you work. Many times, you cannot find a commercial program that does just what you need. Maybe the type of program you need just doesn't exist. If you can write the program you need, you'll meet the needs of your students and increase the efficiency of limited computer time.

GETTING STARTED

Now that you're all fired up and ready to write your own software, let's get started. First of all, don't expect to write a program on your first try that will teach reading to 30 students and keep track of their progress. That is the kind of project that you'll have to work up to.

On the other hand, don't be discouraged by the fact that your first programs seem so insignificant. Simple programs are useful for several reasons.

First, they teach you computer basics. A more complicated program might overwhelm you and cause you to give up in disgust. As you become more sophisticated in your programming, you can come back to the simple program and add features to make it more useful.

Second, simple programs are useful for introducing young people to the computer. When first using the computer, young people need to learn the skills of typing a reply and pressing the ENTER key. It's easier to learn these skills when the expected replys are short and simple.

We have made up programs involving simple addition facts, such as 1 + 2, for kindergarten students. The kindergarten students were challenged by the program. After they left the room, a class of sixth graders entered the room. The simple program was still in the computer. I noticed some of the sixth graders lining up to try out the program. The program was no challenge to them, but they just wanted to use the computer. Don't feel that a program is useless just because it's simple.

This chapter is for two groups of people: those who know some BASIC programming and those who know none. If you know some programming, you can use the explanations and programs in this chapter as a starting point for writing your own software. Perhaps you've learned elements of programming but are unsure about how to combine the elements into an organized program. Maybe you're a good programmer but don't know what types of educational programs are appropriate.

If you don't know programming, this chapter can still be useful to you. One of the frustrating things about having a new computer can be that you don't know what to do with it. Perhaps you bought little or no software and you don't know how to program it. If you're in this situation, you can start by typing the programs in this chapter into the computer. You probably will not understand how the programs work, but you'll have some software to use.

Later, you can continue with the programming chapter and learn BASIC at your leisure. After you learn programming and you go back to the software, you'll understand the earlier programs better. You can then go on to improve these programs and design your own software.

Software production can be divided into three phases. The first phase is the design phase. In this phase, you plan the program. Is this a good candidate for a computer program, or can I achieve my purpose some other way? What exactly will this program accomplish? What is the sequence of this program?

The second phase is the programming phase. Before you actually write the BASIC language program, you need to write out the program in plain English. The more time you spend in the steps so far, the easier it will be to write the BASIC language program.

Phase three involves testing. Usually after you finish writing a program, it doesn't work right. You have to spend time tracking down problems to make the program run properly. Even after the program runs, it still may not do what you want it to. A good way to test a program is to try it out on other people, especially young people. They usually find problems quickly. After they try it out, they can give you suggestions for improving the program. If the problem is minor, return to the programming phase. For major problems, go all the way back to the design phase.

DESIGNING PHASE

A program starts with an idea. The idea may come as a response to a problem. A problem for some fifth graders is that they don't know their multiplication facts. Junior high students have difficulty with outlines.

Next the idea is examined. Not all ideas can be implemented as computer programs. For instance, a program to teach outlining would be very difficult to write. Outlining would probably be better taught by other methods.

Now let's look at the other idea. A program to drill on math facts will be easy to write. Assume for now that the student already knows many multiplication facts and needs practice.

Now we need to state the objectives. The objective for the programmer is to write a program that presents math facts to a student. The objective for the student is to answer a certain number of questions correctly. Let's say that the student is presented with 10 questions and must answer at least eight correctly to pass.

Now that the objectives are stated, the program will be designed in such a manner that the computer can understand it. The type of program design used in this book is called "top down design." This means that the program will be designed in its most general form first. In the case of the drill program, the general outline of the program looks like this:

1. Introduction
2. Drill
3. Final score

So far, the sequence doesn't tell too much about the final program. The details will be worked out later.

After the general outline of the program is given, refinements are made. Where possible, the steps are broken down into substeps. If necessary, the substeps are further divided.

In general, three types of refinements can be made. The three types are sequence, decision, and loop.

Sequence. The sequence refinement involved dividing one step into a number of steps that follow one after the other. When you bake cookies, you follow a recipe. The ingredients are added in a certain order.

In the drill program, the introduction can be subdivided:

A. Print title

B. Print directions

Decision. The decision causes actions to occur if certain conditions are met. In the cooking example, a decision is involved in baking the cookies. After the cookies have baked for about 10 minutes, the cook starts hovering over the oven and checking the cookies. Then comes a decision: If the cookies are done, then pull them out of the oven.

In the drill program, the drill step contains a decision. The computer presents a problem to the student, and the student makes a reply. The computer then "decides" whether the answer is correct. If the answer is correct, the computer gives a positive response, or else the answer is incorrect, and the computer gives a negative response.

Notice the difference in the two types of decisions. In the cooking example, if the condition of doneness was met, then a response was required. This is an "if-then" statement. Writing the statement with indentation makes it easier to understand:

```
If the cookies are done
    Then remove them from the oven
```

The statement from the drill example is an "if-then-else" statement. Two different responses are involved. If the student answer is correct, then response 1 is required. Else response 2 is required. This can be represented:

```
If student response is correct
    Then print positive message
    Else print negative message
```

Loop. Actions that are repeated use loops. When you prepare cookies for baking, you drop dough on the cookie sheet. This action continues until the sheet is full. As long as there's dough, you continue to fill cookie sheets.

In the drill example, the computer continues to pose questions until 10 questions have been asked.

The loop has two parts: the action and the test. The action continues until the test has been satisfied. The test can occur either before or after the action.

Here is a representation of the loop for the math practice program:

```
Continue

     Ask question

     Respond to answer

Until 10 questions have been asked
```

We started out with a general program sequence and then made refinements. Now put the pieces together. Here is the general program outline again:

1. Introduction
2. Drill
3. Final score

This is what the program outline looks like with refinements:

```
Introduction
     Title
     Directions
Drill
     Continue
          Ask question
          Respond to answer
          If answer correct
               Then give positive response
                    Add one to score
               Else give negative response
     Until 10 questions have been asked
Final score
     Print final score
     Ask if student wants to play again
```

Top down program design works well for many types of programs. It has the advantage of showing the general flow of a program first, before getting bogged down in details that can be worked out later. Increasing refinements then further organize the program.

You won't always want to follow the top down approach, however. Sometimes you may get an idea for a program that depends on a computer technique you're unsure of. Before you spend a lot of time on designing the program, you may need to spend some time experimenting at the computer keyboard. If the technique works, then you can go back and finish the design process.

Maybe you'll come up with a completely different software writing procedure. One of the exciting aspects of computer programming is that there's no one correct way to proceed.

PROGRAMMING PHASE

Up until now, the English language has been used. Now the BASIC programming language will be used. Writing a program in a programming language is called "coding" the program.

If the program has been well designed, it'll be easier to code the program. After you've written a few BASIC programs, you'll find yourself thinking in ways that translate into BASIC.

Your program will have two parts. The first part is the main line. The main line of the drill program follows the general sequence. The subroutines will be called from the main line.

The main subroutines in the drill program are the positive and negative response subroutines.

To make programs easier to understand, it's wise to use documentation. Documentation explains how a program functions. There are two forms of documentation: external and internal. External documentation is separate from the program itself.

Part of the external documentation takes place in the design phase. It includes the objectives for the program and the program outline. During programming, another part of the external documentation takes place. That is a description of the variables.

A variable description table will contain the variable name and a short explanation of the variable's function.

Since TI BASIC allows variable names of up to 15 characters in length, the variable name can help to explain itself. As an example, a variable that holds a student's name can be called NAME$.

There is one thing to watch out for, however. Variable names cannot be reserved words. Reserved words are words such as PRINT and GOTO that are part of the BASIC language. The variable name FORREST cannot be used, because FOR is a reserved word.

Internal documentation takes place inside the BASIC program. Several techniques can be used to make the program more understandable. As described above, choosing variable names wisely will reduce confusion as to what a variable represents.

Most important in internal documentation is the use of REM statements. REM statements will help others who use your program to understand what various parts of the program do. Not only that, REM statements will help you when you come back to a program months or years later.

Getting back to the drill program, here is the variable table and program listing:

Variable	*Purpose*
N	number of problems
F1,F2	multiplication factors
ANS	multiplication product
REP$	student reply as a string variable
REP	student reply as a numeric variable
CH	choice variable for positive response
P$	"play again?" variable
R$	"press ENTER when ready variable
L	FOR. . .NEXT loop variable
SC	Number of correct responses
CS	Screen color

```
10 RANDOMIZE
20 REM  MULTIPLICATION DRILLS
30 REM  BY RICHARD MOWE
40 REM  INITIALIZE VARIABLES
50 SC=0
60 CALL CLEAR
70 CALL SCREEN(3)
80 FOR L=1 TO 8
90 CALL COLOR(L,16,1)
100 NEXT L
110 REM  TITLE FRAME
120 CALL CLEAR
130 PRINT " MULTIPLICATION FACTS DRILL":"":"":"":"":""
140 PRINT " PLEASE TYPE CORRECT ANSWER":"":"":""
150 PRINT "  AND PRESS THE ENTER KEY":"":"":"":"":"":"":"":""
160 REM  PRESS ENTER
170 GOSUB 800
180 REM  N = NUMBER OF PROBLEMS
190 FOR N=1 TO 10
200 REM  GENERATE A PROBLEM
210 F1=INT(RND*10)
```

```
220 F2=INT(RND*10)
230 ANS=F1*F2
240 REM  DISPLAY PROBLEM
250 CALL CLEAR
260 CALL SCREEN(6)
270 PRINT F1;"X";F2;"= ?":"":"":"":"":"":"":"":"":"":"":"":""
280 INPUT REP$
290 IF REP$="" THEN 270
300 REM   CONVERT STRING TO NUMBER
310 REP=VAL(REP$)
320 REM   RESPOND TO REPLY
330 IF REP<>ANS THEN 370
340 REM  POSITIVE REINFORCEMENT
350 GOSUB 550
360 GOTO 390
370 REM  NEGATIVE REINFORCEMENT
380 GOSUB 730
390 GOSUB 800
400 NEXT N
410 REM  DISPLAY FINAL SCORE
420 CALL CLEAR
430 CALL SCREEN(7)
440 IF SC<>10 THEN 470
450 PRINT "       YOU SCORED 100%":"":"":"":"":"":"":"":"":""
460 GOTO 480
470 PRINT "  YOU SCORED";SC;"OUT OF";N-1;".":"":"":"":"":"":""
480 PRINT " WOULD YOU OR SOMEONE ELSE":"":"":""
490 PRINT "  LIKE TO PLAY (Y OR N)";
500 INPUT P$
510 IF SEG$(P$,1,1)="Y" THEN 50
520 IF SEG$(P$,1,1)="y" THEN 50
530 CALL CLEAR
540 END
550 REM  POSITIVE REINFORCEMENT
560 REM  ADD 1 TO SCORE
570 SC=SC+1
580 REM  CHOOSE AND PRINT A MESSAGE
590 CH=INT(RND*5)+1
600 CALL CLEAR
610 CALL SCREEN(14)
620 ON CH GOTO 630,650,670,690,710
630 PRINT TAB(8);"YOU GOT IT!":"":"":"":"":"":"":"":"":"":"":""
640 RETURN
650 PRINT TAB(9);"FANTASTIC!":"":"":"":"":"":"":"":"":"":"":""
660 RETURN
670 PRINT TAB(8);"THAT'S RIGHT":"":"":"":"":"":"":"":"":"":"":""
680 RETURN
690 PRINT TAB(10);"TERRIFIC":"":"":"":"":"":"":"":"":"":"":""
700 RETURN
710 PRINT TAB(9);"KEEP IT UP":"":"":"":"":"":"":"":"":"":"":""
720 RETURN
730 REM  NEGATIVE REINFORCEMENT
740 REM  DISPLAY CORRECT ANSWER
750 CALL CLEAR
760 CALL SCREEN(2)
770 PRINT F1;"X";F2;"=";ANS
780 PRINT "":"":"":"":"":"":"":"":"":"":""
790 RETURN
800 REM  PRESS ENTER WHEN READY
810 PRINT "  PRESS ENTER WHEN READY"
820 CALL KEY(0,K,S)
830 IF K=-1 THEN 820
840 RETURN
```

TESTING PHASE

Once the program has been coded, it has to be tested to see if it does what it was designed to do. The first part of the testing phase is called debugging. The term debugging dates back to earlier computers, which had components large enough to be affected by a stray insect. Today, debug means to find the errors that keep a program from functioning properly.

First, find the obvious bugs in the program. When you run a new program, the computer usually will give error messages such as "INCORRECT STATEMENT." If you cannot see the error at first, look closely at the punctuation. A quotation mark either added or missing can cause an error.

Another source of errors is using a variable name that is a reserved word. You just have to scrutinize each variable name in search of reserved words.

After you've corrected errors resulting in error messages, your program still may contain bugs. These bugs will be harder to find. One way to proceed is to correct mistakes in one section of the program at a time. For instance, start with the title page. After it works, move on to a new section.

If you do not know what section of the program contains an error, you can put STOP statements at strategic points to see if the program makes it that far before it bombs. Also helpful are dummy messages like PRINT "I MADE IT" at certain parts of the program to see where the errors are.

If you still cannot find the bugs in your program, show the program to someone else. Often, someone else can see the error you're overlooking.

When your program is debugged, it's time to test it. The best way to test the program is to have people sit down at the computer and run through the program. Eventually, you'll want to test the program on young people of the age and ability of your target group. At first, however, test the program on anyone who will try it. Often, the program faults are glaring to just about anybody but you. That doesn't mean that you're stupid. It's just hard to spot your own mistakes.

Sometimes you'll ask someone to test your program and get the answer, "I don't know anything about computers." That is a good person to test the program. You want to see if the program is easy to use. Ask the tester:

1. Does the computer teach a worthwhile skill?
2. Are the directions clear?

3. Can you easily read the information on the screen?
4. Is the program fun to use?
5. What improvements can I make?
6. Could this task be performed as easily or even more efficiently using some other medium?

After you've tested the program, you should have some changes to make. Evaluate the comments and decide what changes in the program you should make. Some changes are fairly easy and will change only a few lines. Other changes will be more extensive and may be too difficult to do.

Use your judgment. The program doesn't have to be perfect. Do the best job you can with the knowledge you have. You may write a program that others enjoy using but that has flaws. You can come back to the program and continue to improve it as you gain computer sophistication.

USING AND MODIFYING THE PROGRAMS IN THIS BOOK

The next section of the book contains some programs that you can type into your TI computer. There are two purposes for doing this. The first is to give you some software that you can use right away. The other is to teach you.

This is software that you're free to use as you please. If it doesn't do what you want, feel free to change it.

As you type these programs into the computer, you'll learn about programming and about education. At first, you may feel uncomfortable with the computer and feel that you lack computer knowledge.

Don't give up! The beginning of knowledge is a feeling of inadequacy, and the desire for knowledge is thus produced. Doing the programming exercises will extend your computer knowledge. Experience will help you to feel more comfortable with the computer.

The programs presented have been chosen with three criteria in mind. They are meant to be educational, simple to use, and varied.

We already have discussed what makes educational programs. They teach a worthwhile skill. They have clear objectives. They are accurate and use vocabulary and reading levels appropriate for the target audience.

These programs are simple to use. They have been kept short to minimize typing. It's so frustrating to see a good program but realize it'll take you eight hours to type it into the computer.

These programs are easy to understand (at first they may not be) and are well-documented to help you understand program construction.

A compromise was made between educational completeness and simplicity. To produce programs that meet all the criteria would produce difficulties. First, they would take too long to type. Second, they could involve programming techniques beyond the scope of this book.

The plan is to present simple programs that are easy to understand and have some of the good qualities of good educational software. Suggestions will be made which will make the programs even better.

So that you can see the variety of uses for the computer in education, the programs will be varied. Some people think that the computer is used only for math or business. That is not true. You will see computer programs that can be used for a variety of classes. You will be shown how to modify a program to use the same program with different content.

The programs show a variety of programming techniques. The TI has a number of unique features that can be used, such as color graphics.

Finally, programs will be presented that can be used with a variety of age groups. Again, a program can be written for one age group and modified to use with another.

The emphasis, however, will be on the elementary age group. Elementary students are at a good age to use software. Junior high and older students can use software also, but they are getting mature enough to program the computer and make good use of word processing. The programming and word processing worksheets will be best used by junior high and high school students, not to mention adults.

SAMPLE PROGRAMS

These programs can be used in a variety of settings. The first program uses the ARROW ("S" and "D") keys to select correct plurals. The next program tests on the subject of a sentence but can be easily modified for other material. MONSTER MATH is a graphics type addition program. A graphics utility program is then presented that will aid in the display of low-resolution graphics. SPELL PRACTICE drills students on spelling words. Finally, there's a simple but fun drawing program. One last point: The CAP LOCK key will need to be pressed for all of these programs, because when these programs look for a response, they are looking for capital letters only.

Auto Plurals

This is a program that uses the ARROW (FCTN-S and FCTN-D) keys to "drive" a car on the screen to the correct answer.

Objective. Elementary aged children will use this program to increase their knowledge of plurals.

Instructions.

1. RUN the program.
2. Move the car with the FCTN-S or FCTN-D keys.
3. Press the space bar when the car is over the answer.
4. To restart the program, type RUN.
5. To change the plurals, change DATA items starting on line 1150. Each question uses four DATA items:
   - Singular form of word
   - First possible answer
   - Second possible answer
   - Number of correct answer (1 or 2)

Program Outline.

```
Introduction
    Title
    Instructions
    Press the space bar to play
Repeat until all questions are used
    Display information
        Read data
    Show score
    Present question
    Draw car
Respond to answer
```

```
If correct
    Then
        Positive reinforcement
        Add one to score
        Press the space bar to play
    Else
        Give correct answer
        Press the space bar to play
```

Variable Table

Variable	*Purpose*
SNG$	Singular word
PL$(1), PL$(2)	Plural choices
ANS	Number of correct choice
SC	Score
CARSIDE	Location of car (left side of screen = 1, and right side of screen = 2)
MESSAGE	Choice of positive reinforcement
X	Number of spaces car is from left side
L	Loop variable
Y	Vertical screen position for car
K,S	CALL KEY variables

Improvements.

1. Program an ending that will restart the program after each person uses it.
2. Find a way to randomize the data so that the questions will come up in random order.
3. Provide a way to adjust the difficulty of the questions.

```
10 RANDOMIZE
20 REM   AUTO PLURALS
30 REM   BY RICHARD MOWE
40 REM   AND RON MUMMAW
50 CALL CLEAR
60 CALL SCREEN(2)
70 CALL CHAR(128,"0000011F3F3F0C0C")
80 CALL CHAR(129,"00E0F0F8FCFC1818")
90 CALL CHAR(136,"CCCCCCCCCCCCCCCC")
92 CALL COLOR(14,2,5)
```

```
 100 X=1
 110 Y=20
 120 REM  --TITLE PAGE--
 130 CALL CLEAR
 140 RESTORE
 150 CALL SCREEN(2)
 160 PRINT TAB(8);"AUTO PLURALS"
 170 PRINT "":"":"":"":""
 180 PRINT "USE THE S' AND D' KEYS TO"
 190 PRINT "":""
 200 PRINT TAB(7);"DRIVE THE CAR."
 210 PRINT "":""
 220 PRINT " PARK ON THE CORRECT ANSWER"
 230 PRINT "":""
 240 PRINT "   AND PUSH THE SPACEBAR."
 250 PRINT "":"":"":"":""
 260 GOSUB 590
 270 REM  --READ DATA--
 280 READ SNG$
 290 IF SNG$="THATSIT" THEN 1060
 300 READ PL$(1),PL$(2),ANS
 310 REM   --DISPLAY INFO--
 320 CALL CLEAR
 330 PRINT TAB(4);PL$(1);TAB(19);PL$(2)
 340 PRINT
 350 PRINT "WHAT'S THE PLURAL OF ";SNG$;"?"
 360 GOSUB 1010
 370 CALL KEY(0,K,S)
 380 IF K=-1 THEN 370
 390 IF K<>83 THEN 430
 400 IF X=1 THEN 360
 410 X=X-1
 420 GOTO 360
 430 IF K<>68 THEN 470
 440 IF X=29 THEN 360
 450 X=X+1
 460 GOTO 360
 470 IF K<>32 THEN 370
 480 IF X>13 THEN 510
 490 CARSIDE=1
 500 GOTO 530
 510 IF X<17 THEN 370
 520 CARSIDE=2
 530 IF ANS=CARSIDE THEN 560
 540 GOSUB 890
 550 GOTO 270
 560 GOSUB 650
1150 DATA FOX,FOXS,FOXES,2,CHILD,CHILDREN,CHILDS,1
1160 DATA PONY,PONIES,PONYS,1,HILL;HILLES,HILLS,2
1170 DATA MAN,MEN,MANS,1,HOUSE,HOUSS,HOUSES,2
1180 DATA WOMAN,WOMEN,WEMON,1,DEER,DEERS,DEER,2
1190 DATA HAND,HANDES,HANDS,2,MOUSE,MICE,MOUSES,1
1200 DATA THATSIT
 570 GOTO 270
 580 PRINT "":""
 590 REM  --PRESS ENTER TO PLAY--
 600 PRINT "    PRESS ENTER TO PLAY."
 610 CALL SCREEN(4)
 620 CALL KEY(0,K,S)
 630 IF K=-1 THEN 620
 640 RETURN
 650 REM   --POSITIVE FEEDBACK--
```

```
660 CALL SCREEN(2)
670 SC=SC+1
680 MESSAGE=INT(RND*5)+1
690 PRINT "":"":"":""
700 ON MESSAGE GOTO 710,730,750,770,790
710 PRINT TAB(11);"SUPER"
720 GOTO 810
730 PRINT TAB(8);"GOOD ANSWER"
740 GOTO 810
750 PRINT TAB(9);"WAY TO GO"
760 GOTO 810
770 PRINT TAB(9);"KEEP IT UP"
780 GOTO 810
790 PRINT TAB(8);"THAT'S RIGHT"
800 GOTO 810
810 PRINT "":"":"":""
820 PRINT TAB(10);"SCORE=";SC:"":""
830 CALL HCHAR(14,1,136,32)
840 CALL HCHAR(18,1,136,32)
850 CALL SOUND(100,262,10,330,10,392,10)
860 CALL SOUND(1000,330,5,392,5,523,5)
870 GOSUB 590
880 RETURN
890 REM  --NEGATIVE FEEDBACK--
900 CALL SCREEN(2)
910 PRINT "":"":"":""
920 PRINT TAB(9);"** SORRY **":""
930 PRINT "THE PLURAL OF ";SNG$
940 PRINT "":"IS ";PL$(ANS);"."
950 PRINT "":""
960 PRINT TAB(10);"SCORE=";SC
970 PRINT "":""
980 GOSUB 590
990 RETURN
1000 REM  --PLACE CAR--
1010 CALL HCHAR(Y,X,32)
1020 CALL HCHAR(Y,X+1,128)
1030 CALL HCHAR(Y,X+2,129)
1040 CALL HCHAR(Y,X+3,32)
1050 RETURN
1060 CALL SCREEN(2)
1070 CALL CLEAR
1080 PRINT "YOUR FINAL SCORE WAS ";SC;"."
1090 FOR L=1 TO 12
1100 PRINT
1110 NEXT L
1120 CALL SOUND(500,247,10,349,10,392,10)
1130 CALL SOUND(1000,262,10,330,10,392,10)
1140 REM  --PLURAL DATA--
```

■ Make and Take

This is a program that will test a student on the subject of a sentence. A parent or teacher can make up a new test by changing DATA statements.

Objective. Make a computerized test for students to take.

Instructions.

1. RUN the program
2. Student types name.
3. Student types subject of each sentence.
4. Student writes down his score.
5. To end the program, type END when name is requested.
6. To change the test, replace the instructions DATA statement in line 3010. Next replace the questions and answers starting in line 3030. Type the question, a comma, and the answer.

The problem you'll have making up your own test is having a word partially on one line and partially on the line below. To avoid this problem, use the display layout form in Appendix A. Write your question on a copy of the layout form. Find the last word you wrote on the first line. Count the spaces left to the end of the line. Continue the question on the next line.

When you type the question into a DATA statement, type the first line from the layout form. (It will take up one computer line and part of another.) Next, type the number of spaces you counted on the layout form to fill out the line on paper.

Notice line 3010 on the program listing:

```
3030 DATA THE COW JUMPED OVER     THE MOON.,COW
```

There are 5 spaces between the words "over" and "the." This ensures that the word "moon" will appear all on the second line instead of part on the first line and part on the second line.

Program Outline. The general outline for MAKE AND TAKE looks like this:

```
Initialize variables and dimension arrays
Print title
Input name
    If name = "END"
        Then end
        Else give test
```

Below is the outline for the "give test" section.

```
Read Data
```

```
          Instructions
          Questions
          Answers
      Repeat until no more questions
          Print questions centered on screen
          Respond to answer
              If correct
                  Then
                      Print "CORRECT"
                      Add 1 to score
                  Else print "INCORRECT"
      Give final score
```

Variable Table

Variable	*Purpose*
Q$	Question
A$	Answer
NAME$	Name
R$	Reply to question
SC	Score
QNUM	Number of questions
L,N	Index variables
INST$	Instructions
D$	Get variable
ST$	String to be printed in the center of screen
NL	Determines New Line location for string centering routine
Y	Vertical position for string centering routine
X	Horizontal position for string centering routine
A	ASCII code for letter to be printed
S,K	Used in CALL KEY command

Improvements.

1. Randomize order of questions.
2. Store questions, answers, and scores on disk or tape.

```
5 RANDOMIZE
10 REM   MAKE AND TAKE
20 REM   BY RICHARD MOWE
30 REM  INITIALIZE VARIABLES
40 QNUM=10
50 DIM Q$(20)
60 DIM A$(20)
70 REM   PRINT TITLE
80 CALL CLEAR
90 PRINT TAB(8);"MAKE AND TAKE"
100 PRINT "":""
120 PRINT TAB(8);"BY DICK MOWE"
122 PRINT "":"":"":"":"":"":"":"":"":"":"":"":"":"":""
130 GOSUB 650
140 REM  ASK FOR A NAME
150 CALL CLEAR
160 PRINT "WHAT IS YOUR NAME";
170 INPUT NAME$
180 IF NAME$="END" THEN 800
190 IF NAME$="" THEN 150
200 REM   READ DATA
210 READ INST$
220 S=0
230 FOR N=1 TO QNUM
240 READ Q$(N),A$(N)
250 NEXT N
260 REM   GREETINGS AND INSTRUCTIONS
270 CALL CLEAR
280 PRINT "Hi, ";NAME$
290 PRINT "":"":""
300 PRINT INST$
310 PRINT "":"":"":"":"":"":"":"":"":"":"":"":"":""
320 GOSUB 650
330 REM  ASK QUESTIONS
340 FOR N=1 TO QNUM
350 CALL CLEAR
352 ST$=Q$(N)
360 GOSUB 1000
390 INPUT R$
400 IF R$="" THEN 350
410 IF R$<>A$(N)THEN 430
420 GOSUB 540
422 GOTO 440
430 GOSUB 600
432 CALL CLEAR
440 NEXT N
450 CALL CLEAR
460 ST$=" YOU SCORED "&STR$(SC)&" CORRECT."
470 GOSUB 1000
480 GOSUB 650
490 RESTORE
500 CALL CLEAR
510 GOTO 150
530 REM  RESPONSE TO CORRECT ANSWER
540 CALL CLEAR
550 ST$="     THAT IS CORRECT!"
560 GOSUB 1000
570 SC=SC+1
574 GOSUB 650
580 RETURN
590 REM  RESPONSE TO INCORRECT ANSWER
600 CALL CLEAR
```

```
602 NL=-1
610 ST$=" "&R$&" IS INCORRECT!"
612 GOSUB 1000
614 NL=2
616 ST$=" "&A$(N)&" IS THE SUBJECT."
620 GOSUB 1000
622 NL=0
630 GOSUB 650
640 RETURN
650 REM   SUBROUTINE PRESS RETURN TO CONTINUE
660 PRINT "  PRESS ENTER TO CONTINUE";
670 CALL KEY(0,K,S)
671 GOTO 680
672 CH=INT(RND*4)+5
674 CO=INT(RND*15)+1
676 BK=2
677 IF CO=BK THEN 674
678 CALL COLOR(CH,CO,BK)
680 IF K=-1 THEN 670
682 FOR K=5 TO 8
683 CALL COLOR(K,2,4)
684 NEXT K
690 RETURN
800 CALL CLEAR
810 END
1000 FOR L=1 TO LEN(ST$)
1002 Y=12+NL
1004 X=L+2
1010 IF L<29 THEN 1030
1012 X=L-26
1020 Y=13
1030 A=ASC(SEG$(ST$,L,1))
1040 CALL HCHAR(Y,X,A)
1050 NEXT L
1060 RETURN
3000 REM   TEST INSTRUCTIONS
3010 DATA Type the simple subject for        each sentence.
3020 REM  QUESTIONS AND ANSWERS
3030 DATA The cow jumped over the     moon.,COW
3040 DATA Sally went to a movie with  her friend.,SALLY
3050 DATA The sleek red racing car    circled the course.,CAR
3060 DATA Bill found a colorful marbleon the way to school.,BILL
3070 DATA I am being held prisoner in a computer.,I
3080 DATA The computer can do many    things.,COMPUTER
3090 DATA Jack went up the hill with  Jill.,JACK
3100 DATA Lancaster is a wonderful    city.,LANCASTER
3110 DATA The students looked for verbs in the sentences.,STUDENTS
3120 DATA This office is a disgusting mess.,OFFICE
```

■ Monster Math

This is an addition drill with graphics. With minor changes, it can be used for multiplication.

Objective. Primary aged students will use this program to drill on addition facts.

Instructions.

1. RUN the program.
2. Answer addition problems to move on the trail toward home.
3. If you miss a problem, the monster moves toward your home.
4. You win by answering enough problems to reach home.

Program Outline.

```
Introduction
    Create characters for graphics
    Title
    Instructions
    Press return to continue
    Draw playing field
    Initialize variables
Play game
    Input and respond until correct = 5 or incorrect = 3
        If correct
            Then
                Add to your path
            Else
                Give correct answer
                Add to monster's path
                Press return to continue
  Ending
      If number correct = 5
          Then print "YOU LIVE"
          Else print "OUCH"
 Loop back for another game
```

Variable Table

Variable	*Purpose*
OK	Number correct
BAD	Number incorrect
RANGE	Maximum value of addends
A,B	Addends
SUM	Sum of A and B
TRY$	Trial answer as a string variable
TRY	Trial answer as a numeric variable
DLAY	Time delay variable
D$,IN$	GET variables
R	Range of addends
CM$	Color code for monster trail sections
CP$	Color code for player trail sections
L	Loop variable
CH	ASCII code of various graphics characters
CH$	String used to create "block" graphics character

Explanation. This program uses a very handy technique for creating graphics characters that meet your needs. The TI contains codes for all of the uppercase and lowercase letters, various punctuation signs, numerals 0 to 9, and a cursor. Occasionally, you may have a need for other characters, as we did in MONSTER MATH.

Each character that you see on the screen is composed of little "dots" called pixels, which stands for picture elements. The characters are created on a grid 8 pixels across by 8 pixels high. When you first turn on your TI, characters coded 31 through 127 already have been defined. You can either redefine those characters or define characters 128 through 159 to meet your needs.

The method for defining characters is really quite simple. The first step would be to create your character on an 8 × 8 grid, such as shown in figure 7-1.

Step two—determine the code for your character using the dot row chart (figure 7-2).

Each row of the character consists of eight dots. However, each row of our chart only has four dots. Therefore, the "code" for each row of the character will consist of two codes: the left four dot code followed by the right four dot code. Our example with its codes is shown in figure7-3.

FIGURE 7-1

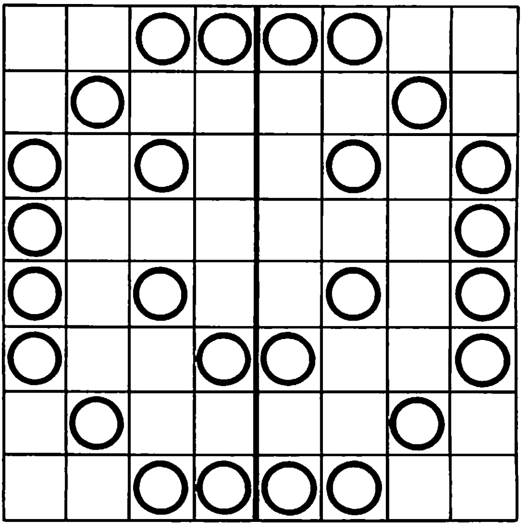
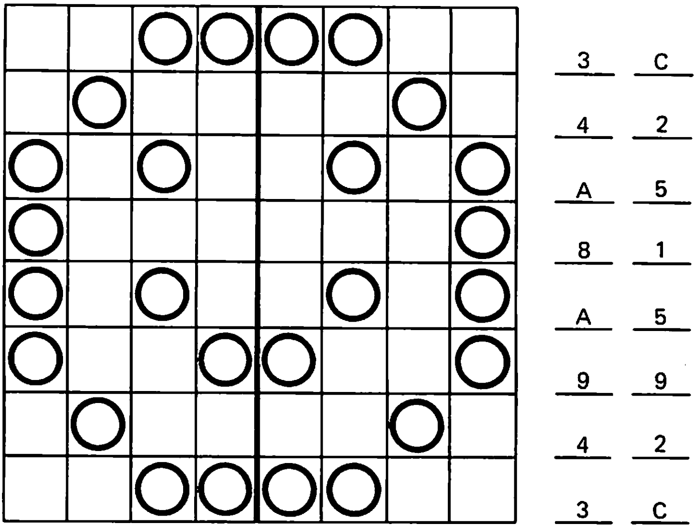


FIGURE 7-2

Ø				
1				○
2			○	
3			○	○
4		○		
5		○		○
6		○	○	
7		○	○	○
8	○			
9	○			○
A	○		○	
B	○		○	○
C	○	○		
D	○	○		○
E	○	○	○	
F	○	○	○	○

FIGURE 7-3



Step 3 would be to choose a character number to use such as 128, the first undefined character. Finally, you tell the computer this information using the CALL CHAR(CN,"----------------") command. CN represents the character number and the dashes in quotes represent the dot code you calculated.

Those of you who understand binary and hexadecimal numbers should be able to see a recognizable pattern to the dot row codes.

Once your character is defined, you can place it on the screen using the CALL HCHAR, CALL VCHAR, or PRINT CHR$(---) commands. The character number you choose will remain the way you defined it (even if it's one of the codes 31 to 127), until you redefine it, or the program ends. If several $'s (character number 36) are on the screen, and your program redefines that character as a stickman, for example, all of the $'s currently appearing on the screen will turn in to stickmen.

Improvements.

1. Make this a game between two players.
2. Provide an exit option.
3. Provide a way for a player to choose to be drilled on addition or multiplication.

```
10 RANDOMIZE
20 REM  MONSTER MATH
30 REM  BY RICHARD MOWE
40 REM  TITLE PAGE AND INSTRUCTIONS
```

```
50 CALL CLEAR
60 GOSUB 1430
70 CALL SCREEN(7)
80 CALL SOUND(750,131,15)
90 PRINT TAB(3);"M O N S T E R  M A T H"
100 PRINT "":"":"":""
110 CALL SOUND(250,147,15)
120 PRINT TAB(5);"TRY TO RUN HOME BY"
130 PRINT "":""
140 CALL SOUND(500,156,15)
150 PRINT TAB(5);"ANSWERING PROBLEMS"
160 PRINT "":""
170 CALL SOUND(500,131,15)
180 PRINT TAB(5);"IF YOU ANSWER WRONG"
190 PRINT "":"":""
200 CALL SOUND(1000,185,15,156,15)
210 PRINT TAB(11);"BEWARE"
220 PRINT "":"":"":"":""
230 GOSUB 1370
240 CALL CLEAR
250 CALL SCREEN(3)
260 REM  DRAW PLAYING FIELD
270 OK=0
280 BAD=0
290 GOSUB 690
300 REM  GENERATE AND PRESENT PROBLEM
310 RANGE=9
320 A=INT(RND*RANGE)+1
330 B=INT(RND*RANGE)+1
340 SUM=A+B
350 PRINT TAB(8);A;"+";B;"=";
360 INPUT TRY$
370 IF TRY$<>"" THEN 470
380 CALL CLEAR
390 GOSUB 690
400 FOR L=1 TO BAD
410 ON L GOSUB 1260,1280
420 NEXT L
430 FOR L=1 TO OK
440 ON L GOSUB 1110,1130,1150,1170
450 NEXT L
460 GOTO 350
470 TRY=VAL(TRY$)
480 IF SUM<>TRY THEN 520
490 CALL CLEAR
500 GOSUB 650
510 GOTO 530
520 GOSUB 580
530 REM   IF GAME OVER THEN RESTART
540 IF OK=5 THEN 240
550 IF BAD=3 THEN 240
560 GOSUB 1370
570 GOTO 320
580 REM   REPLY TO INCORRECT ANSWER
590 CALL CLEAR
600 BAD=BAD+1
610 GOSUB 690
620 IF BAD=3 THEN 640
630 PRINT TAB(8);A;"+";B;"=";SUM
640 RETURN
650 REM   REPLY TO CORRECT ANSWER
660 OK=OK+1
```

```
670 GOSUB 690
680 RETURN
690 REM  PERSON
700 CALL COLOR(14,16,1)
710 CALL HCHAR(4,25,137)
720 CALL HCHAR(4,26,138)
730 CALL HCHAR(5,25,139)
740 CALL HCHAR(5,26,140)
750 CALL HCHAR(6,25,141)
760 CALL HCHAR(6,26,142)
770 REM  MONSTER
780 CALL COLOR(13,2,1)
790 CH=129
800 FOR L=7 TO 9
810 CALL HCHAR(L,2,CH)
820 CALL HCHAR(L,3,CH+1)
830 CH=CH+2
840 NEXT L
850 REM  HOUSE
860 CALL COLOR(15,11,1)
870 CH=145
880 FOR Y=17 TO 18
890 FOR L=15 TO 17
900 CALL HCHAR(Y,L,CH)
910 CH=CH+1
920 NEXT L
930 NEXT Y
940 REM  THE TRAIL
950 CALL HCHAR(11,2,144,6)
960 CALL HCHAR(17,7,144,6)
970 CALL HCHAR(13,13,144,4)
980 CALL HCHAR(9,16,144,8)
990 CALL VCHAR(11,7,144,7)
1000 CALL VCHAR(13,13,144,6)
1010 CALL VCHAR(9,16,144,5)
1020 CALL VCHAR(5,23,144,5)
1030 FOR L=1 TO BAD
1040 ON L GOSUB 1260,1280,1300
1050 NEXT L
1060 FOR L=1 TO OK
1070 ON L GOSUB 1110,1130,1150,1170,11
1080 NEXT L
1090 RETURN
1100 REM  "YOUR PATH"
1110 CALL VCHAR(5,23,136,5)
1120 RETURN
1130 CALL HCHAR(9,16,136,7)
1140 RETURN
1150 CALL VCHAR(10,16,136,4)
1160 RETURN
1170 CALL HCHAR(13,13,136,3)
1180 RETURN
1190 CALL VCHAR(14,13,136,5)
1200 PRINT TAB(7);"** YOU LIVE **"
1210 GOSUB 1730
1220 FOR DLAY=1 TO 1000
1230 NEXT DLAY
1240 RETURN
1250 REM  "MONSTER'S PATH"
1260 CALL HCHAR(11,2,128,6)
1270 RETURN
1280 CALL VCHAR(12,7,128,6)
```

```
1290 RETURN
1300 CALL HCHAR(17,8,128,6)
1310 PRINT TAB(8);A;"+";B;"=";SUM
1320 PRINT TAB(9);"** OUCH **"
1340 FOR DLAY=1 TO 1000
1350 NEXT DLAY
1360 RETURN
1370 PRINT TAB(3);"PRESS ENTER TO CONTINUE.";
1380 CALL KEY(0,K,J)
1390 IF K=-1 THEN 1380
1400 CALL HCHAR(24,1,32,28)
1410 CALL HCHAR(23,1,32,28)
1420 RETURN
1430 CALL CHAR(129,"030C304040888482")
1440 CALL CHAR(130,"C0300C0202112141")
1450 CALL CHAR(131,"8E8E808080809999")
1460 CALL CHAR(132,"7171010101019999")
1470 CALL CHAR(133,"868680404030 0C03")
1480 CALL CHAR(134,"61610102020C30C0")
1490 CALL CHAR(137,"030C304040808C9E")
1500 CALL CHAR(138,"C0300C0202013179")
1510 CALL CHAR(139,"9E8C818183809898")
1520 CALL CHAR(140,"79318181C1011919")
1530 CALL CHAR(141,"8787804040300C03")
1540 CALL CHAR(142,"E1E10102020C30C0")
1550 CALL CHAR(145,"000000070A112040")
1560 CALL CHAR(146,"000000FE01008040")
1570 CALL CHAR(147,"0000000000804020")
1580 CALL CHAR(148,"80809C9C9C9C9CFF")
1590 CALL CHAR(149,"3F202322222320FF")
1600 CALL CHAR(150,"F008C84848C808F8")
1610 CH$="FFFFFFFFFFFFFFFF"
1620 CALL CHAR(128,CH$)
1630 CALL CHAR(136,CH$)
1640 CALL CHAR(144,CH$)
1650 CALL CHAR(152,CH$)
1660 RETURN
1730 CALL SOUND(125,196,15)
1740 CALL SOUND(125,262,15)
1750 CALL SOUND(125,330,15)
1760 CALL SOUND(250,392,15)
1770 CALL SOUND(125,330,15)
1780 CALL SOUND(875,262,15,330,15,392,15)
1790 RETURN
```

■ LOGRAPH

This program will help you write low-resolution graphics programs without using the words CALL HCHAR and CALL VCHAR each time you make a command. The program listing starts with number 6000, so that you can use it as a subroutine in a BASIC program if you want.

Objective. The objective is to provide a program that will simplify the entering of low-resolution graphics data into the computer.

Instructions.

1. Draw the figure on a low-resolution graphics sheet.
2. Set the screen color in line 6000.
3. Find the coordinates for the vertical lines. Enter the coordinates and color in a DATA statement starting in line 9000.
4. Next, enter the coordinates for the horizontal lines in line 9100.
5. If you use this program as a subroutine, line 6340 should be RETURN.

Variable Table

Variable	*Purpose*
CH$	"BLOCK" character
V(1-3)	Vertical coordinates
H(1-3)	Horizontal coordinates
C	Line color
LE	Length of line
DU	Dummy variable for swapping procedure
L	Loop variable
CH	ASCII code for character

Program Outline.

Introduction

    Set memory location variables

    Set screen colors

Read and draw

    Vertical lines

    Horizontal lines

Data

    Screen and frame colors

    Vertical

    Horizontal

    Points

NOTE: Certain characters are redefined in line 6050. Therefore, not all text characters can be used. Also, the colors of character sets 2 to 16 are set in lines 6090 to 6110. Therefore, if characters in those sets are used by your program, their colors will be those set in lines 6090 to 6110, and not what you may expect.

```
6000 CALL SCREEN(2)
6010 CALL CLEAR
6020 REM   GRAPHICS ROUTINE
6030 REM  BY RICHARD MOWE
6040 REM  AND RON MUMMAW
6050 CH$="FFFFFFFFFFFFFFFF"
6060 FOR L=40 TO 152 STEP 8
6070 CALL CHAR(L,CH$)
6080 NEXT L
6090 FOR L=2 TO 16
6100 CALL COLOR(L,L,1)
6110 NEXT L
6120 READ V(1)
6130 IF V(1)=99 THEN 6230
6140 READ V(2),V(3),C
6150 LE=ABS(V(2)-V(1))+1
6160 IF V(2)>V(1)THEN 6200
6170 DU=V(1)
6180 V(1)=V(2)
6190 V(2)=DU
6200 CH=(C-2)*8+40
6210 CALL VCHAR(V(1),V(3),CH,LE)
6220 GOTO 6120
6230 READ H(1)
6240 IF H(1)=99 THEN 6340
6250 READ H(2),H(3),C
6260 LE=ABS(H(2)-H(1))+1
6270 IF H(2)>H(1)THEN 6310
6280 DU=H(1)
6290 H(1)=H(2)
6300 H(2)=DU
6310 CH=(C-2)*8+40
6320 CALL HCHAR(H(3),H(1),CH,LE)
6330 GOTO 6230
6340 GOTO 6340
9000 REM  VERTICAL DATA
9099 DATA 99
9100 REM  HORIZONTAL DATA
9199 DATA 99
```

■ LOGRAPH + WOW

This program shows what you can do with LOGRAPH. It is the same as LOGRAPH, except that DATA statements have been added. It will print the word "WOW" in large letters.

```
6000 CALL SCREEN(2)
6010 CALL CLEAR
6020 REM   GRAPHICS ROUTINE
6030 REM  BY RICHARD MOWE
6040 REM  AND RON MUMMAW
6050 CH$="FFFFFFFFFFFFFFFF"
```

5. The cursor moves forward 24 lines.
6. The cursor moves back 24 lines.

The ROLL UP and ROLL DOWN commands allow you to move through the text quickly. Since the TI displays 24 lines, these commands shift the screen display either one screen page up or down.

7. The cursor moves one tab setting to the right.
8. The cursor moves one tab setting to the left.

These are the TAB and BACK TAB commands. Unless you specify otherwise, the tabs are set at columns 5, 10, 15, 25, 35, 45, 55, and 65.

9. The cursor moves to the next paragraph.
10. The cursor moves to the previous paragraph.

If you want to look through the document one paragraph at a time, then the NEXT PARAGRAPH command will help. The LAST PARAGRAPH command allows you to search backward one paragraph at a time.

11. The cursor moves one line up.
12. The cursor moves one line down.
13. The cursor moves one character to the right.
14. The cursor moves one character to the left.

These are the arrow keys. You can use these keys with either the CTRL or the FCTN key. The arrow keys are the most useful of the cursor movement keys and the ones you'll use the most. If you need to move a little faster through the text, hold down the key and let the automatic repeat feature help you.

At first, you'll use mostly the arrow keys and the NEXT WINDOW commands to move the cursor. As you type longer documents, the paragraph movement functions will become important. When you type multipaged documents, you'll find more use for ROLL UP and ROLL DOWN.

15. You left TI-WRITER. The file FORMATDOC was erased from the screen, although it's still on the disk. When you exit TI-WRITER, make sure that you save an important file to the disk.

WORD PROCESSING LESSON 3

The main tasks done in editing a manuscript are inserting, deleting, and changing text. The last lesson covered cursor movement. This lesson and the next cover some editing techniques.

the CTRL key and a number gives you the commands on the top row.

As you do this worksheet, you will practice moving the cursor, and you'll be asked how far and in what direction the cursor moved. You may have to try some of the commands several times to see where the cursor moved. Sometimes looking at the line numbers will help you see what is happening.

Experiment with these commands. If you press a key for about two seconds, it repeats. See what happens when you repeat these commands.

□ Discussion

One of the disadvantages of a word processor is that you can't look at large portions of your text at one time. A little of the text appears on the screen. Most of it's hidden in the computer memory. That's one reason that moving the cursor is so important. It brings other portions of your text into view.

1. The screen color changed (if you're using a color television or color monitor).

Why is screen color important? On some televisions, one color is clearer than another. For some people, one color is easier to read than another. Doing word processing is demanding on your eyes. Switch through the screen colors several times and see which color is easiest to read.

2. Hopefully, you see the file "FORMATDOC" somewhere in the CATALOG.

The command SD shows the catalog of the disk. In the catalog, you will see a list of your text files along with the programs that make up the word processor. If you want to load a text file but forgot the name of the file, then use the catalog. Notice that the files in the catalog didn't have the prefix DSK1, yet when you load a file, you have to use that prefix.

3. The file "FORMATDOC" was loaded.

"FORMATDOC" is one of the practice text files on the TI-WRITER disk. It's a long enough file to practice the cursor movement commands.

4. The cursor moves one window to the right.

The NEXT WINDOW command moves the cursor through the three screen windows and then back to the first window.

longer using the !E! symbol to remind you to press ENTER. Just remember to press ENTER at the end of each command. If you're typing text, you'll press ENTER if you want to end a paragraph, go to a new line, or skip a line.

Next, the TI-WRITER overlay can help you. As you use the FCTN and CTRL keys, watch to see what the overlay says about the particular key combinations. Pressing the FCTN key and a number key gives you the commands on the bottom row of the overlay. Pressing

WORD PROCESSING WORKSHEET 2

INSTRUCTIONS	QUESTIONS
(LOAD TI-WRITER) (PRESS CTRL 3 FIVE TIMES)	1. How is the display changed? ____
SD 1 (PRESS ENTER AFTER SEEING CATALOG)	2. Do you see "FORMATDOC"? ____
FCTN 9 LF DSK1.FORMATDOC	3. What happened? ____
(PRESS THE KEYS BELOW SEVERAL TIMES)	How far and in what direction does the cursor move when the following keys are pressed?
FCTN 5 (PRESS 3 TIMES)	4. FCTN 5 ____
FCTN 4	5. FCTN 4 ____
FCTN 6	6. FCTN 6 ____
FCTN 7	7. FCTN 7 ____
CTRL T	8. CTRL T ____
CTRL 4	9. CTRL 4 ____
CTRL 6	10. CTRL 6 ____
CTRL E	11. CTRL E ____
CTRL X	12. CTRL X ____
CTRL D	13. CTRL D ____
CTRL S	14. CTRL S ____
FCTN 9 Q E	15. What happened? ____

7. The disk drive started spinning and the disk light came on for a few seconds.

While in the command mode, one of the functions you can perform is to save a file to the disk. This is what you did. SF is the abbreviation for save file. You were prompted with "SAVE FILE, enter filename." On the TI, a file name consists of two parts. The first part is the name of the device that will store the file. In this case, the device name is DSK1 (disk drive 1). The second part of the file name is the actual name of the text file. We are calling this file LESSON1. Therefore the complete file name is DSK1.LESSON1.

8. The file is no longer on the screen.

P stands for purge the file. The reason you had to type Y is that the computer is giving you a chance to change your mind. If you type a seven-page report and typed P by mistake, the computer gives you a chance to reconsider before you wipe out your work.

9. The file you erased is back on the screen.

LF is the load file command. Again, you were prompted to enter the file name. This time, the file name was displayed on the screen. The computer remembers the last file name you used for a file command. In this case, you were using the same file name as before, so you pressed ENTER, and the computer assumed you wanted to load the same file you had saved. If you had wanted to load a different file, then you would have typed the name of the new file right over the old file name. If any of the old file name was still showing, you would have pressed the SPACE BAR to get rid of the rest of the old file name.

10. Hopefully, the paragraph you typed was printed on the printer.
11. Except for the first letter of each paragraph, the other letters should have been printed in lowercase.

See, the printer really does print lowercase letters!

WORD PROCESSING LESSON 2

The first lesson gave you a general idea of what a word processor can do. This lesson will start examining TI-WRITER in more depth. You'll concentrate on ways to move the cursor throughout the text. Since you can't see large portions of text at one time as you can with paper, it's important to be able to move the cursor quickly through the document to find your place.

Before you start Worksheet 2, let's discuss few things. First, I'm no commands you can use in working with TI-WRITER.

Below the command line is the cursor. The cursor is the blinking rectangle that shows your place in the document or on the command line.

3. The number 0001 is displayed.

Each line has a number. Certain commands use the line numbers to operate. For instance, if you want to delete lines from the file, you have to specify the line numbers of the lines to delete.

4. The last words are END OF FILE.

When you pressed E !E!, you left the command mode and entered the edit mode. The command line disappeared from the screen, and you were able to start entering text on the screen.

As you typed the practice paragraph, did you notice anything strange? After you typed all the way across the screen, a new screen appeared. As you typed to the edge of the second screen, a third screen appeared. When you filled up the third screen, the first screen appeared again and you were typing on the second line. The reason that the screen keep flipping back and forth is that the computer displays an 80-character line. The TI screen can display only 40 characters, but a normal sheet of paper usually displays up to 80 characters. By switching screens, TI-WRITER gives you an idea of how the finished document will look.

5. The Carriage Return (CR) symbol appears, and the cursor drops to the next line.

When you use the word processor, you don't have to press ENTER at the end of every line, as in BASIC. You press ENTER for the following reasons:

To end a paragraph

To end a line and print on the next line

To skip a line.

When you typed the short paragraph, did you remember to capitalize? Remember, lowercase letters appear on the TI as small uppercase letters, and uppercase letters appear larger. If the ALPHA LOCK key is up, then you capitalize by using the SHIFT key as on a typewriter.

6. Pressing FCTN 9 put you back into the command mode.

Notice that the command line again appears at the top of the screen.

WORD PROCESSING WORKSHEET 1

INSTRUCTIONS	QUESTIONS
(INSERT TI-WRITER CARTRIDGE AND DISK) (TURN ON PERIPHERALS,TV,AND CONSOLE) (PRESS ANY KEY TO BEGIN) (PRESS 2 FOR TI-WRITER) (PRESS 1 FOR TEXT EDITOR)	1. Did you have to press ENTER?
	2. What are the first 3 words on the screen? ____ 3. What number is displayed?____ 4. What are the last words on the screen? ____
E! E!	
(TYPE THIS PARAGRAPH) Typing with the TI-WRITER word processor is fun. Look at those crazy characters on the screen. !E!	5. What happened when you pressed ENTER? ____
FCTN 9 SF !E! DSK1.LESSON1 !E!	6. What happened? ____ 7. What happened? ____
FCTN 9 P !E! Y !E!	8. Is the paragraph you typed still on the screen? ____
FCTN 9 LF !E! !E!	9. What happened? ____
(TURN ON PRINTER) FCTN 9 PF !E! (TYPE DEVICE NAME AND PRESS ENTER)	10. What happened? ____ 11. Were upper- and lowercase letters printed? ____
(TURN OFF COMPUTER)	

As you complete the worksheets, you'll have two types of instructions. They are instructions to do and words or commands to type.

Instructions to do are typed in parentheses. The first instruction on worksheet 1 is INSERT TI-WRITER CARTRIDGE AND DISK. This is an action that you perform.

When you use the word processor, you'll type mostly text and commands. When you see these words, this is what to type:

!E! Press the ENTER key.
CTRL S Hold down the CTRL key and the S key at the same time.
FCTN 9 Press the FCTN key and the 9 key at the same time.

Text will be shown with no parentheses. It will be typed on the worksheet in upper- and lowercase letters in paragraph form.

To summarize the two types of instructions:

To do (INSERT THE DISK)
To type Hey, this is fun.

Here are a few hints.

1. Make sure the ALPHA LOCK key is up.
2. If you stall on the worksheet, look ahead to the Discussion section. Then go back over the worksheet.
3. If you're away from the computer for longer than nine minutes, and the screen goes blank, press CTRL S to get the picture back.
4. DON'T BE AFRAID TO EXPERIMENT.

□ Discussion

The first section of the worksheet helps you get TI-WRITER up and running.

1. You didn't have to press ENTER.

Sometimes the computer asks for a response and waits for you to press ENTER. Other times it goes into action as soon as you press a key. If you're not sure whether to press ENTER after giving an answer, just watch the computer screen. If nothing happens, then press ENTER.

2. The first three words are EDIT, TABS, and FILE.

The top line is the command mode prompt line. It lists the

USING THE LESSONS

In writing these lessons, I'm assuming that you have only one disk drive. If you have more that one drive, you'll have to make slight changes to the worksheets. The TI-WRITER manual explains multi-disk drive operation.

Before starting let's take care of a few details. First, you'll need to make a backup copy of the program disk. Then if anything happens to the original, you'll have a copy.

Not only that, but it simplifies your use of TI-WRITER. Normally, you use a separate disk for the TI-WRITER program and your text files. However, when you're just beginning, you can store your text files on the program disk. Then you don't have to keep switching disks.

When you type longer text files, you probably will want to use separate data disks for storing test files.

To make a backup disk, you'll need the disk manager command module, TI-WRITER program diskette, and a blank disk (or one you want to erase and reuse). Then follow the instructions on pages 16 and 17 of the disk memory system manual. If you don't understand the instructions, then find someone to help you.

The next detail concerns your printer. When you print a file, you'll have to know the device name for your printer. If you're using the TI 99/4 printer, the device name is RS232 for Lesson 1 and RS232.LF for Lesson 5. For more information on using the TI printer as well as other printers, read pages 118 to 121 in the TI-WRITER manual.

Finally, find the TI-WRITER overlay (the strip of plastic with the orange and gray dots). The overlay will show you which keys to press to use special functions.

These lessons are meant to acquaint you with TI-WRITER software. The only way to learn to use a word processor is to get your hands on the keys and use it. Worksheets are included to take you through the important aspects of the word processor. The worksheets are not meant to take the place of the operating manual, however. Keep the manual handy to answer questions and learn more about TI-WRITER. The more you learn, the more you'll want to know.

WORD PROCESSING LESSON 1

The first lesson is an introduction to the word processor. You'll practice loading the TI-WRITER disk and enter a short paragraph. Then you'll save the text file on the disk.

word processor more efficiently. Even primary age children enjoy dictating stories to an older person and seeing the finished printed product.

USING THE TI AS A WORD PROCESSOR

To use your 99/4A for word processing, you'll need the proper hardware and software.

Hardware

To use your TI as a word processor, certain hardware is required. First, you need the TI 99/4A console. The TI 99/4 console isn't compatible with TI-WRITER.

Then you need the peripheral expansion system with the memory expansion card, disk controller card, and RS 232 interface card.

The next consideration is the disk memory drive. You'll need at least one. It may be either an internal or external drive. Two disk drives make certain operations more convenient. You can insert the word processing software in drive 1 and your data disk, which contains your text files, in drive 2. If you have only one drive, you'll have to switch disks once in a while, which is usually not too inconvenient.

For the video display, you can use a television. If you do a lot of word processing, you might want to consider buying a monitor. The letters are easier to read and cause less eyestrain.

Your word processing needs will dictate the choice of printer. If you want typewriter-quality printing, you'll need a letter-quality printer. If you need readable copy, then a dot-matrix printer is satisfactory. The printer must be compatible with the RS 232 interface.

One of the most important pieces of advice you'll get on hardware is to *make sure all the pieces work together.* If you don't understand how to set up and test the system with TI-WRITER, then get help.

Software

To do the worksheets in this chapter, you'll need the TI-WRITER word processor. This program includes a solid-state cartridge and a program diskette. You'll need both of these along with the overlay for the keyboard.

On the computer, it's not so simple. Instead of moving the carriage, you move the cursor. To do this, you need to learn special commands. Not only that, but you have to figure out where to move the cursor. Different commands will move the cursor in different directions various amounts.

The problem isn't that you can't find specific places in the text file. The problem is that you have to use new methods that take time to learn. If you don't use the word processor often enough, you won't get the practice required to be proficient. Then it may be faster and less frustrating to use a typewriter.

LEARNING WORD PROCESSING

Word processing is a powerful tool that will be used in the future by increasing numbers of people. Some questions need to be considered. First, what should its place be in the school curriculum?

There are several reasons to teach word processing. The first is that it is a job skill. Many students who are training to be secretaries and typists will use word processors on the job. Another reason to teach word processing is that it is an aid to composition. Since word processing makes editing so easy, it could be used to help students write better papers. A third reason to teach word processing is that it contributes to computer literacy. It shows an important way that computers are used and it gives students a glimpse of what writing will be like more and more in the future.

To get the maximum benefit from a word processor, you need to touch type. It would be useful to offer word processing as part of a typing class. Another possibility would be to offer typing one semester and word processing the next. Since computers are more expensive than typewriters, it may be more feasible to have one or two word processors in the typing class. Students could rotate between the typewriters and word processors.

The next question is, who should study word processing? Anyone who is going to do a lot of writing or typing would profit from studying word processing. Of course, the first priority should be to learn typing. If you can't study word processing now but will need it later, take a typing class now so you'll be ready to study word processing later.

This does not mean that you can't learn word processing if you don't know how to type. Even elementary school students can learn to use a word processor. As a form of computer literacy, it's important to show young people word processing. Later they can learn to use the

You can compose sentences on the screen. If they don't look good, you can do that part over, even if you're in the middle of the paper. Sometime you may type something you know will need to be changed later. Since it's easy to edit, you can relax.

If you want to later retrieve and edit a text file, disk storage is an important advantage of word processing. A teacher could use the word processor to make up a new grade report form for parents. Over time, the form could be revised, because it is stored on the disk.

Another application that makes use of the retrieval feature is called boilerplating. An example is a contract. A contract may require special paragraphs that are the same for each contract. Since the paragraph is on file, it can be retrieved with a few simple commands and inserted in the new document.

For some applications, the ability to make multiple copies is important. A copy machine can make multiple copies, but the word processor can make multiple copies with minor changes in content or formatting.

DISADVANTAGES OF WORD PROCESSING

Even though word processing has substantial advantages over using a typewriter, the word processor is not the best choice for every application. Word processing has some disadvantages that deserve consideration.

The first problem is expense. Compare the cost of a word processor and a typewriter. Even if you have a computer already, you may have to buy expensive additions to your system. Even then, if you use a dot-matrix printer, the printing will not match the quality of a typewriter.

Another problem, if you don't do a lot of typing, is the complication of learning to use the word processor. Some of the complications have been mentioned. There is the problem of an all uppercase display on the TI, and the fact that the screen display does not give a true indication of how the printed page will look.

Another problem concerns moving the cursor. When you use a typewriter, you insert the paper and start typing. You can see where you're typing on the paper. If you need to type something in a special place on the paper, you move the carriage to the proper place and start typing. If after typing part of a page, you need to return to the top of the page and correct an error, you move the carriage to the proper place.

You'll notice another difference between the TI and the typewriter. A sheet of paper is usually 8-1/2 inches wide. If you use 1-inch margins on the paper, that means that you can type 65 characters across the page. It takes 54 lines of typing to fill the page. You can see the entire page at one time.

On the TI screen, you can type 40 or less characters. At this point, a new portion of the screen appears and part of your typing disappears. Type to the right margin again to see the third part of the screen. Altogether, you can type 80 characters across the screen. You just won't be able to see all of them at one time. Twenty-four lines fill the TI screen. You can type more, but the first lines will roll off the top of the screen. (Don't worry. They're not lost.) When you type on the TI, what you see on the screen will be different from what the printer prints out. Before you give up in disgust over the complexities of word processors, realize this. If you do a lot of writing, you'll quickly learn to overcome the obstacles mentioned. (More about word processor disadvantages later.)

As you type, the letters are displayed on the screen. Later you'll learn ways to correct your mistakes right on the screen. If you need to delete a word, the words to the right will shift left to close up the gap. If you need to insert an extra word, the words to the right will move out of the way.

After you've typed and edited the text to your satisfaction, a few simple commands will save your text file on the disk. Other commands will print the file on paper.

ADVANTAGES OF WORD PROCESSING

Depending on your application, word processing offers important advantages over the typewriter. These advantages are ease of editing and formatting, disk storage, and the ability to make multiple copies.

The main advantage of word processing over the typewriter is ease of editing and formatting. Since the text changes involve only the mistakes and not a complete retyping, time is saved. In most cases, the saved time translates into money. Other times it means increased productivity.

For anyone doing much writing, the word processor can decrease anxiety associated with writing. When you use the typewriter, typing errors can be very frustrating, especially if you're under pressure to meet a deadline. You want the paper to be just right and cringe with every mistake.

The word processor can decrease the anxiety of making mistakes.

Using a word processor, the author types the manuscript. Using a printout of the text, the author makes corrections on the paper. Next comes the neat part. Instead of typing the complete manuscript over, he retypes only the corrections. After the editing, a new manuscript is printed by the computer. If there are still mistakes, only the corrections are retyped.

After the manuscript is finished, the author retains a disk that contains the book manuscript. If he later writes a revision of the book, much of the book may be retrieved from the disk.

Advertisers use word processors. Aren't you thrilled to receive a letter with your name and address neatly typed in at the appropriate place? It's probably the work of a word processor. The advertiser can use a standard letter form. Every letter is the same except for the addressee's name and address. When the letters are printed, the computer reads from two files, the text file, and an address file.

Secretaries find many uses for word processors. Since some of their correspondence is repetitious they can keep a file of standard letters, When it's time to send a letter, they can sometimes find one in their word processing files. With a few changes, the letter is ready to print with a savings in time.

The word processor has vast potential for teachers. If you're a teacher, you know the time you have to spend at the typewriter. Some of that time could be saved if you didn't have to completely retype a test that needed to be revised or retype a spirit master that wore out. As word processors become less expensive and more readily available in schools, teachers will find innovative uses for the word processor.

HOW ARE WORD PROCESSORS USED?

To use a word processor, you first plug in the TI-WRITER module and insert the word processing disk. The screen will prompt you with various options. First choose the TI-WRITER option and then the text editor option. Then you'll find yourself in the word processing program. Type E (for Edit) and press the ENTER key.

Now you're ready to type. You'll notice some differences between typing on a TI and on a typewriter. First, the TI displays in all capital letters. Usually you type in lowercase letters. The way the word processor handles the problem is by interpreting the normal capital letters on the screen as lowercase letters. To capitalize the letter, you first press the SHIFT key. On the screen, the "capitalized" letter will appear as a larger capital letter.

you'll need to correct mistakes (unless you're perfect). Word processing allows you to go back through the text and correct any errors. You do this without retyping all of the text. You only retype the mistakes. If you find that you'd like to switch the fourth and tenth paragraphs, a few simple commands will accomplish the move.

When you have the text as you want it, you then store the text on a floppy disk. The information is stored in a text file. This text file later can be loaded from the disk into the computer for further editing or printing. The final step in the word processing cycle is printing. The software allows you to format the text depending on your needs. For instance, you may need two copies of a paper. The first copy will be printed double spaced with wide margins. The second copy will be printed single spaced with narrow margins.

To use word processing, you need a word processor. A word processor has two main parts: hardware and software. The hardware is the computer itself. Many offices use word processors that are special purpose computers that do only word processing. They are called dedicated word processors, because word processing is all that they do. They cannot be programmed for other purposes.

It's very expensive to buy a computer and use it only for word processing. For most people, it makes more sense to use a computer for a number of applications. Your TI computer can be used as a word processor. You'll need the TI 99/4A (not 99/4) console, disk memory system (memory drive and controller), memory expansion, RS 232 interface, and an RS 232 compatible printer. With your computer, you'll be able to do word processing and still have other capabilities.

The other component of a word processing system is software. Word processing software has instructions on a floppy disk and module that turn your computer into a word processor. Here, when the term "word processor" is used, it means a TI 99/4A computer using word processing software.

WHO USES WORD PROCESSORS?

Word processors are useful for tasks that involve a lot of writing. For the writer, editing is normally a painful process. The author usually first types a rough draft. Next comes the editing. Whole sections may be added, deleted, or moved. Some sections have to be rewritten. After the editing is complete, the manuscript is rewritten, a monumental task, if you type only 30 words a minute, as I do. The revised manuscript still probably contains mistakes, some of which cause more retyping.

chapter 9 USING A WORD PROCESSOR

One of the main strengths of the computer is its versatility. The same computer, with the aid of proper software, can be used to process payroll checks or to compute complicated scientific formulas. It can both entertain and instruct. It can make you a more productive thinker.

Computers were first developed to deal with numbers. The first widely used computer language was called FORTRAN, which stands for "Formula Translator." Eventually, however, it was discovered that computers can handle information skillfully in the form of words as well as numbers.

One of the ways information is manipulated is by a process called word processing. The first section of this chapter explains what word processing is and what a word processor does. The second section shows how to use word processing software. Worksheets are included to guide you through word processing using TI-WRITER.

WHAT IS WORD PROCESSING?

Word processing is a means to type, edit, store, and print text. The first step involves typing information into a computer. As you type, the words appear on the computer screen. After you type the information, Another part of the diagnostic section will analyze individual keystrokes to pinpoint your weakest keys. Then the practice part of the diagnostic section will let you practice the weak individual keystrokes.

The game section of TOUCH TYPING TUTOR provides practice in typing words and number/symbol combinations. As the game proceeds, the difficulty level increases, thus motivating the player to type faster.

TOUCH TYPING TUTOR provides practice for typists of all ability levels. The game, however, is set for an initial typing speed of 15 words per minute (wpm). This target wpm can be lowered to 10, but even the lower speed may make the game too frustrating for beginning typists.

The other sections of the program use the target wpm as a way of assessing progress, so that a slower typing speed won't hinder use of the program.

Even though typists of all ages can use this program, adults and students seventh grade and older will find TOUCH TYPING TUTOR to be most useful.

software. As you type, you know immediately your speed and accuracy. It becomes a game to try to improve. In fact, some of the typing software comes in a game format. You'll find yourself typing faster than you thought possible.

On the negative side, the constant feedback can be a disadvantage. Some people feel threatened when told they're wrong. The noises that signal mistakes can slow these people down. When they see their slower typing speed displayed, they get even more discouraged.

Another disadvantage is that the typing software can't teach you proper typing technique. It can drill you on the home row (asdfghjkl), but it doesn't know if you're holding your fingers correctly and using the right fingers on the right keys.

Some typing software will help you type faster but won't help you develop the format for writing a business letter. To learn the proper format for a term paper, you'll have to look elsewhere.

A good way to use typing software to your advantage is to use it as a supplement. It can give you good practice while you learn the basics in a typing class. If the typing class uses computers, the typing programs could form part of the class instruction. Typing software also can be used for practice once you already know how to type. It's easy to become rusty at typing if you don't use the skill for awhile.

If you're a parent working with your son or daughter, software can help you. Remember, however, it won't do the job without your help. A high school typing class meets five periods a week for one semester (about 18 weeks). For you to accomplish the same amount of work that is covered in a one-semester course will call for a considerable time investment from you and your child.

You might consider buying a typing book. Then decide what is important, and use the book along with the typing software.

USING TOUCH TYPING TUTOR

The TOUCH TYPING TUTOR module has three sections: lessons for beginners, a diagnostic section for the experienced typist, and a game that most will enjoy.

The lessons teach finger placement on the letter, number, and symbol keys. The program provides drill in the form of single keystrokes, nonsense syllables, and words. Periodically, the lessons review keys from previous lessons.

The diagnostic section tests your typing skill and gives practice in weak areas. You can find your typing speed in words per minute.

Just make sure that you don't push. Otherwise they might become discouraged.

HOW TO LEARN

There are two aspects of learning to type. The first is learning the keys and being able to find and strike them quickly. That's what most people think about when they think of learning to type.

The second aspect, which is more important, is being able to type something worthwhile. For instance, to type a letter, you need to know how to use proper letter form, depending on the type of the letter. If you're typing a term paper, you need to know something about the proper format for the term paper.

For most people, the best way to learn to type is to take a class. Typing takes discipline. You have to sit at the typewriter and refrain from looking at the keys or the paper. Then you start out typing inspiring words such as "fjdksl." It's not a natural thing for a person to teach himself.

Learning to type takes a lot of repetition. It's something that you have to keep at. There are times when it would be easy to give up. If you've learned to hunt and peck and want to learn to touch type, it's frustrating and slow to have to type "fjdksla;" over and over. It's easier to have a teacher to encourage you.

Finally, it helps to have a teacher look over your shoulder from time to time and correct bad typing habits. You may think your style is more comfortable or will help you type faster, when in the long run, it will only slow you down.

One question comes up, though. Will it hurt to learn to type on a typewriter and then switch to a computer? After all, the computer and typewriter keyboards aren't the same. Not only that, but on a computer you usually have a different typing style, more numbers, and symbols. You don't even type on paper; you type on a screen.

Sometimes switching between the typewriter and the computer is a problem, but only a temporary one. Most of the keys are the same. Once you get used to the differences, you'll do fine.

Some of you will want to learn to touch type but are not able to take a typing class. You may have noticed that there's typing software on the market and wondered if you could teach yourself to type.

Typing software has some advantages and disadvantages. Using typing software is motivating. The programs are fast paced and will hold your attention. Monitoring is one of the motivating features of the

There's a mystique about the computer. Learning to master the computer gives a real feeling of power, which is important for people who feel unsuccessful in other areas.

Many students feel discouraged because of poor handwriting. Typing, therefore, contributes to self-confidence because it produces neat printing, whether on paper or on the computer screen.

Finally, anyone who writes a lot, whether on the typewriter or word processor, should learn to type.

WHEN TO LEARN TYPING

At what age should children learn to type? This isn't a new question. Typewriters have been around for quite awhile. Some children become fascinated with the typewriter at an early age and learn to type. For the most part, however, students learn to type when they take a typing class.

With the increasing use of computers in homes and schools, younger and younger children are using computers. Many children learn to type by the "hunt and peck" method rather than learning to "touch" type. This may cause problems for them later. Generally, they won't learn to type as fast or as accurately as others and the bad habits they pick up will be difficult to break later on.

One option is to teach typing in elementary school. The problem with that option is that elementary school students usually have not developed the coordination needed to learn touch typing. They might give up in frustration and want nothing to do with typing when they're old enough to learn properly.

Another option is to postpone their use of computers until they're in junior high and can learn to touch type. Obviously, that isn't going to happen.

So what is the solution? First of all, realize that not all young people are going to learn to touch type. Many people are happy to hunt and peck and see no need to learn touch typing.

Second, don't push. Typing classes are usually first offered in the seventh to ninth grades. That's soon enough for formal instruction. Elementary students can do a lot on the computer without learning to touch type. It's a time for exploration. Typing training can come later.

For elementary students, don't emphasize speed. That can encourage the development of bad typing habits. Let them develop their own style. Of course, some elementary students have the coordination to learn touch typing. For those students, it wouldn't hurt to teach them.

age. Many children start using a computer before they enter kindergarten.

Another reason to examine the teaching of typing is that more people will need to learn to type. Many more jobs exist that use the computer. Even cash registers are becoming more sophisticated and have complicated keyboards. Another question arises. Is it important to take a typing class, or can a person learn to type on his own?

The computer keyboard is a little different from a typewriter, and the style of typing used in computer applications is often different. If you do BASIC programming, you have to type a lot of numbers and symbols. Does it matter if you learn to type on a computer or a typewriter first?

Software exists that will drill you on typing while keeping track of your speed and errors. Is this method of learning typing as good as taking a class?

It's important to find answers to the above questions. This chapter examines the learning of typing and gives you some suggestions both for learning and improving typing skills.

WHY LEARN TO TYPE?

The traditional reasons for learning to type are still valid today. The computer has added to these reasons.

As in the past, typing is an important job skill. Secretaries will continue to be in demand. They will need to know how to type, in many cases, on either a computer or typewriter. Even typewriters are becoming more computerized and have memories and other features. Other office workers will be typing on computers or other business machines with computer-like keyboards.

Besides the traditional typing jobs, new careers are opening up that use the computer. Students in school now will enter jobs that don't even exist today. Any student with an interest in computers should learn to type.

Students preparing for college also will want to learn to type. Typewritten papers aren't any better written because they're typed, but they're easier to read and tend to get higher grades. As word processing systems become more readily available and inexpensive, learning to use the computer keyboard will increase in importance.

Another reason to learn to type is to make more efficient use of the computer. Learning a skill such as typing can contribute to self-confidence, because it's an important step in mastering the computer.

chapter 8 LEARNING TO TYPE

Over the years, some subjects become more important to learn, while other subjects become less important. One hundred years ago, children learned to ride horses as a means of transportation. Today riding horses is a hobby.

Ten years ago, the skill of using a computer wasn't widely taught below the college level. Computers were expensive enough that most people couldn't afford them. The development of the microcomputer changed all that.

Now that microcomputers can be purchased for less than a hundred dollars, their use is increasing rapidly. In a few years, the home computer will be as much a part of many households as the television. Suddenly, it's important to know something about computers.

One computer skill has been with us since before the invention of the computer. That's typing. Traditionally, most typing students have taken typing to prepare for college or a secretarial career. They usually start to learn to type in junior high or high school.

With the increasing use of microcomputers, it's important to reexamine the skill of typing, for several reasons. The most important reason is that more and more children are learning to type at an earlier

```
1060 IF Y<25 THEN 1080
1070 Y=24
1080 GOTO 200
1100 REM  COLOR
1110 C=C+1
1120 IF C<16 THEN 1140
1130 C=2
1140 GOTO 200
1200 REM  COLOR DOWN
1210 C=C-1
1220 IF C>1 THEN 1240
1230 C=15
1240 GOTO 200
```

```
290 IF K=46 THEN 900
292 IF K=90 THEN 1000
294 IF K=67 THEN 1100
296 IF K=86 THEN 1200
298 GOTO 200
300 REM  UP
302 CALL HCHAR(Y,X,C*8+32)
310 Y=Y-1
320 IF Y>0 THEN 330
322 Y=1
330 GOTO 200
400 REM  DOWN
410 CALL HCHAR(Y,X,C*8+32)
420 Y=Y+1
430 IF Y<25 THEN 450
440 Y=24
450 GOTO 200
500 REM  LEFT
510 CALL HCHAR(Y,X,C*8+32)
520 X=X-1
530 IF X>0 THEN 550
540 X=1
550 GOTO 200
600 REM  RIGHT
610 CALL HCHAR(Y,X,C*8+32)
620 X=X+1
630 IF X<33 THEN 650
640 X=32
650 GOTO 200
700 REM  UPPER LEFT
710 CALL HCHAR(Y,X,C*8+32)
720 X=X-1
730 Y=Y-1
740 IF X>0 THEN 760
750 X=1
760 IF Y>0 THEN 780
770 Y=1
780 GOTO 200
800 REM  UPPER RIGHT
810 CALL HCHAR(Y,X,C*8+32)
820 Y=Y-1
830 X=X+1
840 IF X<33 THEN 860
850 X=32
860 IF Y>0 THEN 880
870 Y=1
880 GOTO 200
900 REM  LOWER RIGHT
910 CALL HCHAR(Y,X,C*8+32)
920 X=X+1
930 Y=Y+1
940 IF X<33 THEN 960
950 X=32
960 IF Y<25 THEN 980
970 Y=24
980 GOTO 200
1000 REM  LOWER LEFT
1010 CALL HCHAR(Y,X,C*8+32)
1020 X=X-1
1030 Y=Y+1
1040 IF X>0 THEN 1060
1050 X=1
```

```
            Go to input routine
    If input is C or V
        Then
            Increment color value
            Go to input routine
Flash CURSOR if no input
```

Variable Table

Variable	*Purpose*
X, Y	Horizontal and vertical locations
K,S	CALL KEY variables
CH	61 or C*8+32 for CURSOR character change
C	Current color
L	Loop variable

Improvements.

1. Provide for a color option that doesn't change the display as the CURSOR is moved about the screen.
2. Provide an option for saving the screen once it's completed.

```
10 CALL CLEAR
12 CALL SCREEN(2)
20 X=14
30 Y=12
40 C=2
50 CH=61
100 FOR L=40 TO 152 STEP 8
110 CALL CHAR(L,"FFFFFFFFFFFFFFFF")
120 NEXT L
130 FOR L=2 TO 15
140 CALL COLOR(L,L,1)
150 NEXT L
200 CALL KEY(0,K,S)
201 IF CH=61 THEN 205
202 CH=61
203 GOTO 210
205 CH=C*8+32
210 CALL HCHAR(Y,X,CH)
220 IF K=-1 THEN 200
230 IF K=69 THEN 300
240 IF K=88 THEN 400
250 IF K=83 THEN 500
260 IF K=68 THEN 600
270 IF K=49 THEN 700
280 IF K=61 THEN 800
```

■ Sketch

This program provides you with a utility for drawing pictures on the screen using low-resolution color graphics.

Objective. To allow children to create colorful screen graphic pictures using a minimum of keystrokes.

Instructions.

1. RUN the program.
2. Set the color by pressing the C to go up in color value, or V to decrease the color value (if you set the color to black, the CURSOR will remain white).
3. Create a picture by moving the CURSOR using the following keys:

   E — up
   S — left
   D — right
   X — down
   1 — up and left
   = — up and right
   Z — down and left
   . — down and right

This program creates a block character 8 pixels high by 8 pixels wide. Fourteen of these characters are created, each in a different character set, so that they each may be a different color.

NOTE: In order to erase, set the color to black(#1).

Program Outline.

```
Initialize variables and screen
Input routine
    Depending on input, go to proper subroutine
        If input is CURSOR movement key
            Then
                Set block at old X,Y
                Update X or Y
```

```
750 CALL HCHAR(23,1,32,32)
760 CALL HCHAR(24,1,32,32)
770 ST$=" SORRY, MY WORD WAS............. "&WD$(J)&"!"
780 NL=5
790 GOSUB 1240
800 GOTO 940
810 REM  TOO LATE REPLY
820 CALL SCREEN(15)
830 CALL CLEAR
840 PRINT "TOO LATE.  MY WORD WAS......";WD$(J);"."
850 PRINT "":"":"":"":"":""
860 GOTO 940
870 REM  CORRECT ANSWER REPLY
880 CALL CLEAR
890 CALL SCREEN(12)
900 PRINT "YEA!! YOU GUESSED IT!"
910 PRINT "":"":"":"":"":""
920 SC=SC+PT*LEN(GU$)
930 REM  SHOW SCORE
940 ST$=" YOUR SCORE = "&STR$(INT(SC))&"."
950 NL=8
960 GOSUB 1240
970 GOSUB 1180
980 LE=1
990 GU$=""
1000 NEXT J
1010 CALL CLEAR
1020 PRINT "YOUR FINAL SCORE WAS";INT(SC);"%."
1030 END
1040 REM  INITIALIZE VARIABLES
1050 RESTORE
1060 RANDOMIZE
1070 R=10
1080 FOR L=1 TO R
1090 N=INT(RND*10)+1
1100 IF WD$(N)<>"" THEN 1090
1110 READ WD$(N)
1120 TT=TT+LEN(WD$(N))-1
1130 NEXT L
1140 SC=0
1150 LE=1
1160 PT=100/TT
1170 RETURN
1180 PRINT "  PRESS ENTER TO CONTINUE."
1190 CALL KEY(0,K,S)
1200 IF K=-1 THEN 1190
1210 IF K  13 THEN 1190
1220 RETURN
1230 REM  SCREEN LOCATING ROUTINE
1240 FOR L=1 TO LEN(ST$)
1250 X=L
1260 Y=12+NL
1270 IF X 33 THEN 1300
1280 Y=Y+1
1290 X=L-32
1300 CH=ASC(SEG$(ST$,L,1))
1310 CALL HCHAR(Y,X,CH)
1320 NEXT L
1330 RETURN
1340 DATA HOUSE,COMPUTER,ROBOT,VACATION,STICKS,POLICE,DANCER,HUMAN,
     FASCINATION,K
     NIGHT
```

```
130 CALL SCREEN(4)
140 GOSUB 1180
150 CALL SCREEN(2)
160 CALL CLEAR
170 GOSUB 1050
180 PRINT "HERE IS A LIST OF THE WORDS"
190 PRINT "":""
200 LINES=3
210 FOR L=1 TO R/2
220 PRINT WD$(L);TAB(14);WD$(L+R/2)
230 PRINT
240 LINES=LINES+2
250 NEXT L
260 FOR L=1 TO 22-LINES
270 PRINT
280 NEXT L
290 PRINT TAB(3);"PRESS ENTER TO BEGIN."
300 CALL SCREEN(7)
310 GOSUB 1190
320 FOR J=1 TO R
330 CALL SCREEN(2)
340 CALL CLEAR
350 ST$=" PRESS THE SPACEBAR WHEN YOU     KNOW THE WORD."
360 NL=6
370 GOSUB 1240
380 CALL SCREEN(3)
390 DLAY=1
400 LETTER=ASC(SEG$(WD$(J),LE,1))
410 CLUE$=SEG$(WD$(J),1,LE)
420 CALL HCHAR(12,LE+1,LETTER)
430 CALL KEY(0,K,S)
440 IF K=32 THEN 500
450 DLAY=DLAY+1
460 IF DLAY<50 THEN 430
470 LE=LE+1
480 IF LE=LEN(WD$(J))THEN 820
490 GOTO 390
500 CALL SCREEN(6)
510 ST$="FINISH THE WORD, THEN PRESS ENTER."
520 CALL HCHAR(18,1,32,32)
530 CALL HCHAR(19,1,32,32)
540 PRINT ST$;
550 REM  INPUT GUESS
560 CALL KEY(0,K,S)
570 IF K<>32 THEN 590
580 IF GU$="" THEN 560
590 IF K=-1 THEN 560
600 IF K=13 THEN 710
610 IF K<>8 THEN 670
620 IF GU$="" THEN 560
630 GU$=SEG$(GU$,1,LEN(GU$)-1)
640 CALL HCHAR(11,LE+1,32)
650 LE=LE-1
660 GOTO 560
670 GU$=GU$&CHR$(K)
680 LE=LE+1
690 CALL HCHAR(11,LE+1,K)
700 GOTO 560
710 WORD$=CLUE$&GU$
720 IF WORD$=WD$(J)THEN 880
730 REM  ANSWER INCORRECT REPLY
740 CALL SCREEN(8)
```

Variable Table

Variable	*Purpose*
L,J	Loop variables
WD$	Word list
WORD$	Clue given added to guess
DLAY	Timing loop variable
CLUE$	Clue
LE	Length of clue
GU$	Guess
SC	Score
ST$	String to be printed on screen
NL	Distance ST$ will be from center of screen
LETTER	Individual letters of clue
K,S	CALL KEY variables
PT	Point value per letter
LINES	Number of screen lines used
X,Y	Horizontal and vertical coordinates for placing ST$
CH	ASCII code of individual characters of ST$
R	Length of list
N	Random number 1 to 10
TT	Total number of letters that could be guessed

Improvements.

1. Add more sounds.
2. Make it a two-player game with the points going to the player who first presses a designated key.
3. Provide for a series of players, and keep track of their scores by name.

```
10 REM   SPELL PRACTICE
20 REM   BY RON MUMMAW
30 CALL SCREEN(2)
40 CALL CLEAR
50 PRINT "S P E L L  P R A C T I C E"
60 PRINT "":"":"   I WILL SHOW YOU WORDS"
70 PRINT "":"   ONE-LETTER-AT-A-TIME."
80 PRINT "":"":"WHEN YOU THINK YOU KNOW THE"
90 PRINT "":"WORD, PRESS THE SPACEBAR AND"
100 PRINT "":"   COMPLETE THE WORD THAT"
110 PRINT "":TAB(9);"I STARTED."
120 PRINT "":"":"":"":""
```

4. Student writes down his score.
5. To restart the program, type RUN.
6. To change the words, retype the DATA in line 9000.

Since the screen is only 28 spaces wide, words may only be 12 letters or less in length. Otherwise, they won't fit properly on the screen when the initial list is displayed. Larger words could be used if lines 240 to 270 were changed to:

```
240 FOR L = 1 TO R
250 PRINT WD$(L)
260 LINES = LINES + 1
270 NEXT L
```

This change would allow a list of 20 words, each word up to 28 letters long.

Program Outline.

```
Print title and instruction page
Initialize variables
Choose words in a random order
Give test
    Show word one letter at a time
    If answer is correct
        Then
                PRINT "CORRECT"
                Add points to score
    Otherwise
        If no answer
            Then
                PRINT "TOO LATE"
        If wrong answer
            Then
                PRINT correct answer
Show score
Show final score
```

```
6060 FOR L=40 TO 152 STEP 8
6070 CALL CHAR(L,CH$)
6080 NEXT L
6090 FOR L=2 TO 16
6100 CALL COLOR(L,L,1)
6110 NEXT L
6120 READ V(1)
6130 IF V(1)=99 THEN 6230
6140 READ V(2),V(3),C
6150 LE=ABS(V(2)-V(1))+1
6160 IF V(2)>V(1)THEN 6200
6170 DU=V(1)
6180 V(1)=V(2)
6190 V(2)=DU
6200 CH=(C-2)*8+40
6210 CALL VCHAR(V(1),V(3),CH,LE)
6220 GOTO 6120
6230 READ H(1)
6240 IF H(1)=99 THEN 6340
6250 READ H(2),H(3),C
6260 LE=ABS(H(2)-H(1))+1
6270 IF H(2)>H(1)THEN 6310
6280 DU=H(1)
6290 H(1)=H(2)
6300 H(2)=DU
6310 CH=(C-2)*8+40
6320 CALL HCHAR(H(3),H(1),CH,LE)
6330 GOTO 6230
6340 GOTO 6340
9000 REM  VERTICAL DATA
9010 DATA 6,12,5,8,13,13,6,8,14,14,7,8
9012 DATA 9,13,8,8,13,13,10,8,14,14,9,8
9014 DATA 6,12,11,8,7,13,13,5,7,13,19,5
9016 DATA 6,12,21,7,9,13,24,7,6,12,27,7
9018 DATA 13,13,22,7,14,14,23,7,14,14,25,7
9020 DATA 13,13,26,7
9099 DATA 99
9100 REM  HORIZONTAL DATA
9110 DATA 14,18,14,5,14,18,6,5
9199 DATA 99
```

■ Spell Practice

This program gives the user a list of words, one letter at a time, until there are enough letters for the student to complete the word.

Objective. To help the student practice spelling a group of preselected words.

Instructions.

1. RUN the program.
2. Student presses the spacebar when he recognizes a spelling word.
3. Student completes the spelling word and presses ENTER.

In this lesson, you'll have to type more paragraphs. Remember to press ENTER only at the end of the paragraph. Don't be too concerned if the paragraphs have little mistakes. Do the best you can and, as you learn more about TI-WRITER, you can come back and correct the mistakes.

After typing the paragraphs, you'll save the text on disk. This has two benefits. First, it will give you practice saving a file. Second, it will give you a copy of the original paragraphs.

When you do the rest of the worksheet, you may make mistakes and want to redo the paragraphs. Rather than retype them, you can reload the original paragraphs using the LF command.

□ Discussion

Typing these paragraphs will give you further practice in typing at the computer. Hopefully, it's getting easier for you.

1. The line numbers disappeared and the text shifted to the left.

The first part of the worksheet concerns setting margins. The left margin was set at 5 and the right margin at 35. With these margin settings and no line numbers, all the text fits on one screen, so you can see all the text at once.

2. If you typed to the right margin and were in the middle of a word, then the word jumped to the next line at the left margin.

This is called wordwrap. It's an advantage of the word processor over typing. When you type on a typewriter, you have to be careful near the right margin not to exceed it. With the word processor, you just type and let the computer worry about which line to put the word on.

3. SF saves the file to the disk.
4. Yes, the word "use" disappeared.

Sometimes, you need to replace text when you're editing. You just move the cursor over the first part of the document that you want to replace. Then, as you type, the old text disappears.

5. The word "then" was deleted.

Maybe as you type, you find that you've added an extra word in the text. Use the DEL CHAR command to get rid of the unwanted text.

6. The words to the right moved out of the way to make room to insert characters.

FCTN 2 is the INS CHAR command. With the word processor, you

WORD PROCESSING WORKSHEET 3

INSTRUCTIONS	QUESTIONS
(LOAD TI-WRITER) T (MOVE CURSOR WITH SPACE BAR TO 5) (PRESS L BUT NOT “ENTER”) (MOVE CURSOR WITH SPACE BAR TO 35) (PRESS R AND THEN “ENTER”	
FCTN Ø	1. What happened? ______
(TYPE THE FOLLOWING PARAGRAPH) In this lesson, you will learn some editing techniques on the computer. First you will use arrow keys to find the proper place. You will then try commands that change, insert, or delete text.	2. What happened to words typed near the right margin? ______
FCTN 9 SF DSK1.LESSON3	3. What does SF do? ______
(HOLD DOWN CTRL KEY. USE ARROW KEYS TO MOVE CURSOR TO “U” IN “USE”.) try	4. Did the word “use” disappear? ___
(USE CTRL AND ARROW KEYS TO MOVE CURSOR TO “T” IN “THEN”.) (PRESS FCTN 1 FIVE TIMES)	5. What happened to “then”? ______
FCTN 2 eventually (PRESS CTRL 2 AND WATCH CLOSELY)	6. What happened? ______ 7. What happened? ______
FCTN 9 Q E	

can move words out of the way to squeeze in new words. If you fill up the line with text, don't worry. TI-WRITER will give you another blank line.

7. The gap closed up and the words were rearranged on the line.

REFORMAT is a housekeeping command. After you delete or insert words, you need to rearrange the words on the line. REFORMAT will arrange words until the CR symbol is found.

To become familiar with editing, you'll need to practice. The TI-WRITER manual contains many useful editing exercises. The manual also gives more details on the various commands.

WORD PROCESSING LESSON 4

In the last lesson, you worked on inserting, deleting, and changing text. In this lesson, you'll explore other editing techniques.

Sometimes you need to find a particular place in the file. If you know the line number of the spot, there is a command to help you. Maybe you need to find a certain word or phrase. Again TI-WRITER can assist.

Sometimes your paper looks good, but two blocks of text need to be switched. The ability to move sections of text around is one of the strong points of the word processor.

You'll be loading the file PRACTICE from the TI-WRITER disk. This file has some mistakes in it. The reason is that it's a practice file used with the TI-WRITER manual for correcting mistakes.

Some of the commands on this worksheet use line numbers. Be aware of the line numbers as you do the exercises. Again, if you goof badly, just reload the PRACTICE file and start over.

□ Discussion

You started with L, the Lines command. All the Lines commands use line numbers.

1. The Command Line shows MOVE, COPY, DELETE, or SHOW Lines.

The option that you chose was S for Show. When prompted for the line number, you chose 23.

2. The cursor is on line 23.
3. The cursor is on line 1.

WORD PROCESSING WORKSHEET 4

INSTRUCTIONS	QUESTIONS
(LOAD TI-WRITER) LF DSK1.PRACTICE FCTN 9	
L S 23	1. What does the COMMAND LINE say?______________ 2. What line number is the cursor on? ______________
FCTN 9 S 1	3. What line is the cursor on? ______
FCTN 9 SH FS /apparent/ (lowercase letters)	4. What word is the cursor on?______
FCTN 6 FCTN 9 RS /apparent/obvious/ Y	5. What happened to "apparent"? __
FCTN 9 M 14 23 1 (SPACE BETWEEN NUMBERS) FCTN 6 FCTN 5 FCTN 5	6. What happened to the paragraph order?______________
FCTN 9 D 2 23 (SPACE BETWEEN THE NUMBERS)	7. How many lines of the file are left? ______________

Sometimes, you need to move the cursor to the start of the file. You can use the Show command and specify line 1.

4. The cursor is on the word "apparent."

You used Find string, one of the Search commands. You told the computer that you wanted to find the word "apparent." The search progresses through the file, starting at the current cursor position. That's why the cursor was moved to the start of the file.

The command Find string locates the word or phrase and stops. The other Search command is Replace string. It both finds the text and replaces it.

5. The word "apparent" was replaced with "obvious."

The sequence /apparent/obvious/ means search for "apparent" and replace it with "obvious." Once the computer finds "apparent," it gives you several choices. You can decide whether or not to carry out each replacement individually or to have all occurrences of the word "apparent" automatically replaced by "obvious."

6. The two paragraphs switched places.

Isn't that something! You used the Move command. Here is the meaning of the numbers, 14 23 1. The start of the text to be moved is line 14 and the end is line 23. This block of text is being inserted after line number one. The FCTN 6 and FCTN 5 commands move the cursor so you can see what has happened.

7. Only one line (the title) of the file remains.

You used the Delete command. The 2 is the first line deleted, and 23 is the last line deleted.

As usual, you'll have to keep at it to become comfortable using these commands.

WORD PROCESSING LESSON 5

Eventually you'll want to print your text file. One of the strengths of a word processor is that you can print your manuscript in many different formats. You can choose single or double spacing. If you want the right margin to be justified, you can do that, too. That would be very time consuming on a typewriter.

To do this lesson, you'll need a printer to be connected. You'll also need to know the device name of your printer. If you have a TI 99/4 printer, then the device name for this lesson is RS232.LF. For more information, read pages 118 to 121 in the TI-WRITER manual.

WORD PROCESSING WORKSHEET 5

(LOAD TI-WRITER)

(TYPE THESE FORMAT COMMANDS AND THE PARAGRAPH THAT FOLLOWS)

```
.FI  (UPPERCASE)
.AD
.LM 15   (SPACE BETWEEN M AND 1)
.RM 7Ø
.IN +5
.CE
```

THIS IS A REVIEW

In your study of TI-WRITER, you have learned many things about word processing. In the first lesson, you received a brief overview of TI-WRITER. You loaded the program, typed a paragraph, saved and loaded a disk file, and even printed the file.

In Lesson 2, you used the cursor movement keys to find your way around the text. The most useful keys for moving the cursor are the four arrow keys.

In Lesson 3, you learned how to set the margins. You then went on to use commands that change, insert, and delete text.

In the last lesson, you discovered ways to locate specific lines or words in the text. Special commands allowed you to replace, move, and delete portions of the text.

Lesson 5 will show you how to print a file using the TEXT FORMATTER. The commands at the top of this worksheet with periods in front have special significance when this file is printed through the TEXT FORMATTER.

INSTRUCTIONS	QUESTIONS
FCTN 9 PF (ENTER DEVICENAME)	1. Is the file printed the way you typed it?________________
FCTN 9 SF DSK1.LESSON5	(ANSWER THESE QUESTIONS AFTER COMPLETING THE WORKSHEET)
FCTN 9 Q E 2	2. Where is the left margin set? ____ 3. What do you notice about the right margin? ______________ 4. What do you notice about the title?

```
DSK1.LESSON5
(ENTER PRINT DEVICENAME)
(PRESS ENTER)
(PRESS ENTER)
(PRESS ENTER)
(PRESS ENTER)
```

5. Are the paragraphs indented? ___

For this lesson, the format of the worksheet will be slightly different. This is because you'll have more text to type. As you type the five paragraphs, you'll also be reviewing what you've learned using TI-WRITER.

Another difference of this lession is that you'll be using the TEXT FORMATTER program to print your file, which will allow you to use format commands in your text. Format commands set such things as margins, paragraph indentations, and centering. And for a really professional look, you can justify the right margin.

□ Discussion

The most enjoyable part of word processing is watching the finished product roll off the printer. To people unacquainted with word processing, it's like magic.

1. Yes, the file is printed the way you typed it.

Some printers, however, will skip an extra line when they come to the CR symbol.

You can print out a file from the TEXT EDITOR like you just did, to see the file before it's printed from the TEXT FORMATTER.

2. The left margin is set at 15.

This is accomplished by the .LM 15 command.

3. The right margin is justified.

The commands .FI (FILL) and .AD (ADJUST) are used for right justification. You have to use both commands.

4. The title is centered.

The command .CE is used for centering.

5. Yes, the paragraphs are indented.

The command .IN +5 indents the paragraphs.

HELPFUL HINTS

Even though the word processor can help you to be more productive, it doesn't take away your need to think. The techniques of composing with the word processor are different, but a little practice will help you to use the word processor effectively.

You'll first need an idea of what you're going to write. An outline can help you. You can use the word processor to make the outline, or you can use pencil and paper, whichever seems most comfortable. Don't feel that because you have a word processor you have to use it for all phases of writing.

Another way to organize your thinking is with 3×5-inch index cards. Carry them around with you, so that when you get an idea you can write it on the card. When you're ready to type, arrange the cards in order. If it would help, you can use the cards to form an outline.

When you use the word processor, remember that you don't have to worry about mistakes. You can correct them later. Just start typing. Sometimes you'll have difficulty starting. Try typing anything that comes into your head. If you don't like the look of the first paragraph, erase it and start over. Don't be too critical of yourself, however. Remember, you can edit later.

If you make a mistake and catch it right away, go ahead and correct it. Since it's so easy to correct errors and fun to watch the letters moving around on the screen, you'll probably be tempted to do more editing than you should. Save most of the editing for later. Your first task is to get your thoughts on the screen. When you can't think of what to write, your eyes will wander over what you have typed, and that's when you'll want to clean up a phrase here and do some rewording there. Resist the temptation!

As you type, try to stay on the topic, but don't worry if your thoughts aren't completely in order. With the word processor, you can change the order of the paragraphs later. If you think of something that you just have to write that's completely out of order and you're afraid that you'll forget it if you have to come back to it later, then save the text file you are working on to disk. Then type the new file. Save the new file on disk and load the old file from disk and go back to working on it. It's alright to have a number of separate files on the disk. Later you can put them in order and combine them into one file.

After the rough draft is complete, it's time to edit. There are two ways you can proceed. The first is to edit right on the screen. If you have a short paper, you may be able to scroll through the file and find mistakes. For longer papers, there are problems with this method.

First, it's easy to get confused about what is a capital letter and what is lowercase. Another problem is access. You can't see all of the file at once. The most you can see at one time is a few lines.

The other way to edit is to make your corrections on a printout of the text file. This has several advantages. First is portability. You can take the printed paper wherever you want to do the editing. Second is continuity. You can spread out the sheets and see all of the file. Third is format. You can see how the printed page will look. Finally, it gives you a backup. Disks sometimes fail, and it's good to have at least one printed copy.

After the file is edited on paper to your satisfaction, you can do the editing at the computer. Using the search function, you can speed to the places that need correction. At first, you may need to go through the printout and edit cycle several times before you have a copy that meets your standards. Don't get discouraged. You'll improve with practice. Using a word processor is like learning to type. At first it's difficult, but the more you practice, the faster you get.

When you first use the word processor, there will be a lot that you won't understand. You'll use only a small fraction of the word processor's features. As you use it more, you'll start to understand why certain things are done the way they are. Keep the reference card and the manual handy. You'll learn by getting into situations where you need to know something new. Then you'll start digging in the manual until you find out what you need to know.

One thing you'll find, as you do more and more word processing, is that you'll soon accumulate many data disks. You would do well to organize the disks by categories. If you use meaningful titles for the text files, that will further simplify your organizing and help you identify a file six months later.

Since disks are fragile and do fail sometimes, keep backups of your important files. Backups can be in both disk and hard copy form.

Be very careful of certain commands. If you intend to load a file from the disk, but type SF instead of LF, you'll wipe out your file. One precaution to take is to protect your files, using the DISK MANAGER module. Then you can't accidentally erase them. If you have to edit a protected file, unprotect it before you resave it.

If you're a parent, your son or daughter probably will enjoy using the word processor. If your children are too young to read, they'll enjoy dictating stories for you to type. They like seeing their stories in print, even if they can't read them.

chapter 10
SUMMARY

If you've read this far, you're probably amazed at how much computer knowledge you've gained. Like any other skill, you learn to use a computer through persistence and practice.

In order to help you understand and appreciate what you've learned so far, here are some of the main points of the book.

GETTING STARTED

A computer system is divided into two main parts, hardware and software. The computer itself, along with components such as the disk drive and monitor, is the hardware. The programs that run the computer make up the software.

Depending on the age and ability of the young people you work with, you'll choose appropriate computer activities. With younger children, you'll want to use more software. As young people get older, they will increase in their ability to write their own programs and use a word processor.

If you're working with young people and computers, keep a few things in mind. First, encourage balance. Remember that, while computers are fun, there are other worthwhile activities, too. Second, encourage students to experiment with the computer. Part of the fun of using a computer is seeing what it will do.

On the other hand, there are some things to discourage. The main things are impatience and pushing. You may feel that your son or daughter should learn to program a computer by the age of 12, but please don't push too hard.

USING COMMERCIAL SOFTWARE

Four types of software are discussed in this book. The first type is the tutorial. It strives to teach you a new skill. The next type is drill and practice. It gives you an opportunity to practice skills you've been exposed to previously.

Simulation software recreates a situation for you and lets you control the new environment. Simulation is valuable for situations that are dangerous, time consuming, or expensive.

Productivity software allows you to use the computer to become more productive or creative. One example is word processing software.

Beware! Finding the proper software for your needs can be frustrating. These frustrations will be minimized if you can try out the program before you buy it.

When you're looking for educational software, keep in mind the qualities of good software. The most important quality is educational soundness. The program should teach something important and should accomplish its objectives. The reading level of the program should be understandable for the intended user.

Once that quality is achieved, the program should be easy to use. Information on the screen should be formatted so that it's easy to read. When a response is required, the type of response should be obvious. You should be able to move both forward or backward through the program.

To test the interest level of a program, have young people use the program and tell you how they liked it.

A good educational program uses the computer well. Some of the motivational aspects of the TI computer are color graphics, sound (watch out in a classroom), timing, and scorekeeping. If the objectives of the program could be accomplished just as well with pencil and paper, then the program is not using the computer well.

Finally, software should be well-documented. Written documentation should include clear instructions for both the student and teacher. Objectives and lesson plans further increase the usefulness of the program.

If you're looking for good software, here's some advice. First, become knowledgeable. You can do this by taking classes, reading, talking to people, and visiting computer stores. Next, explore sources of public domain software (check Appendix E). As you become more knowledgeable, write as much of your own software as you can. Then buy the software that you can't write yourself or copy legally.

PROGRAMMING

Learning to program can help you in a number of ways. First, it's a valuable skill that can help you find a job. Second, it can help you learn to solve problems logically and systematically. Third, gaining control of the computer can help your self-confidence.

When you program a computer, you use a programming language. This book teaches you two programming languages. BASIC is the language that almost all microcomputers use. LOGO is a good introductory programming language.

WRITING SOFTWARE

If you're tired of paying for commercial software that can't be copied and modified, learn to write your own. The first step in writing software is the design phase. In this phase, you plan the program. You use your goals and objectives to make a program outline.

The next step is the program phase. You translate the program outline into a BASIC language program.

After the program is written, you enter the testing phase. Trying the program yourself and having others try it helps you to work the bugs out of the program.

Finally, there's the documentation phase. Documentation helps you and others understand the program. Internal documentation consists of remark (REM) statements within the program. External documentation is written material explaining the main parts of the program.

LEARNING TO TYPE

As more people use computers, typing will gain importance. For most people, the best way to learn to type correctly is to take a class. Typing software is available to help you practice.

USING A WORD PROCESSOR

A word processor has two main parts: hardware and software. Word processing hardware is the computer, screen, disk drive, and a printer. Word processing software is simply a word processing program contained on a floppy disk and a module.

To use a word processor, type your text into the computer. Then you go over the text and correct your mistakes. Finally, the printer types the text.

CLOSING THOUGHTS

Computers are changing all of our lives. Two keys to coping with the changes are knowledge and balance.

Knowledge is important, because there's a lot of confusion regarding the use of computers in education. If you study and learn to use the computer properly, you can make a lasting contribution to the education of young people.

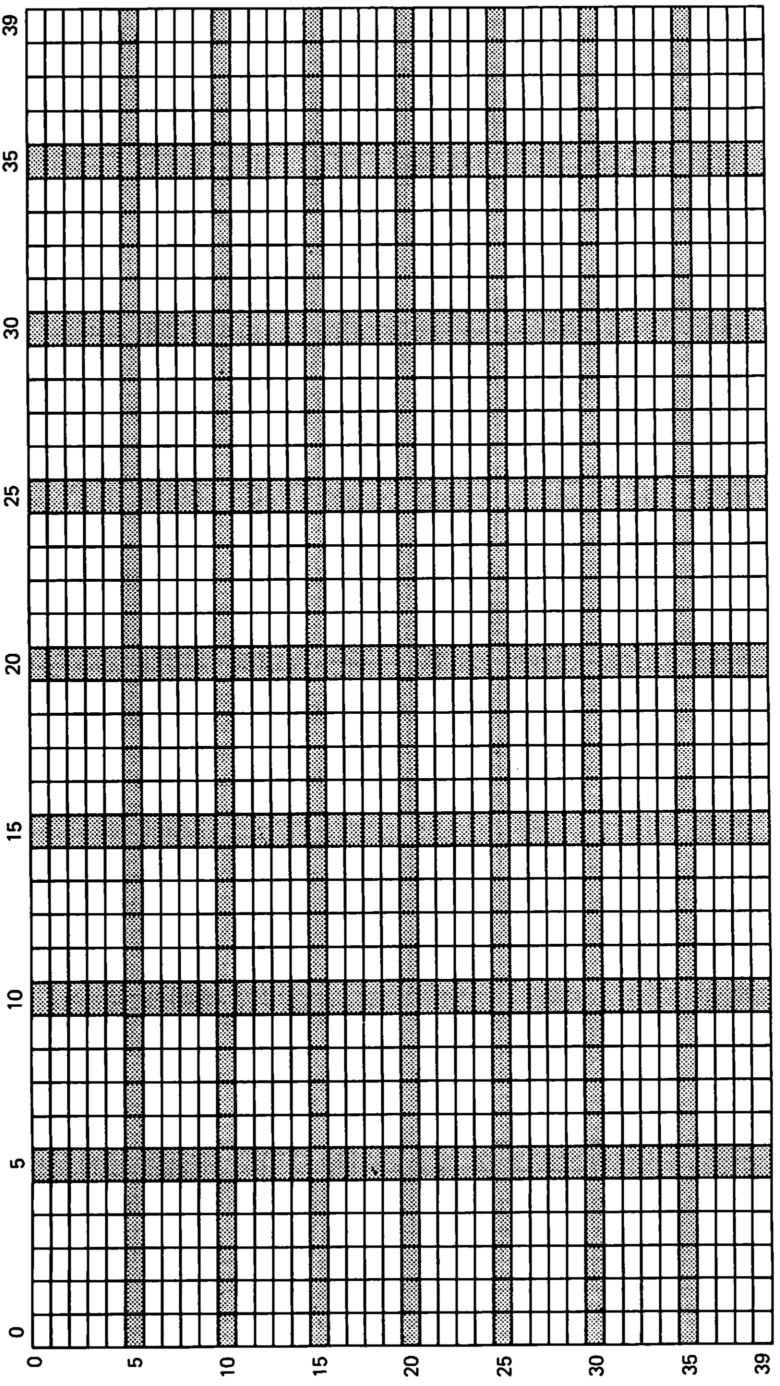
Balance is important, because it's easy to lose your perspective, where computers are concerned. Computers are powerful tools. Like other power tools, they must be used wisely.

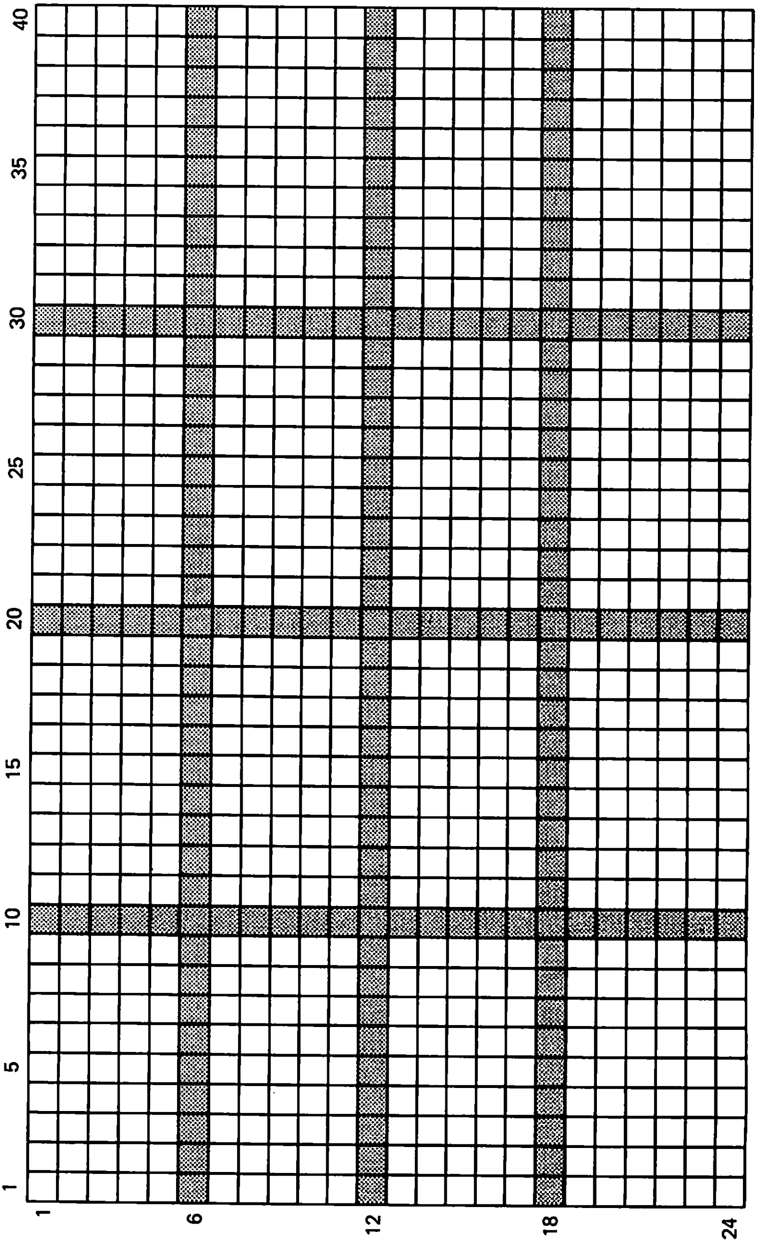
Appendix-A

HELPFUL FORMS

The two helpful forms are:

1. Low-resolution graphics form
2. Text layout form.


0
5
10
15
20
25
30
35
39
0
5
10
15
20
25
30
35
39


1
5
10
15
20
25
30
35
40
1
6
12
18
24

Appendix-B

HELPFUL BOOKS

The Texas Instruments Home Computer Idea Book
by David H. Ahl
Creative Computing Press, 1983

Are You Computer Literate?
by Karen Billings and David Moursund
Dilithium Press, 1978

The Micro Millennium
by Christopher Evans
Washington Square Press, 1979

Teaching BASIC Bit by Bit
Batya Friedman and Twila Slesnick, Editors
University of California, Berkeley, 1980

Using and Programming the TI 99/4A, Including Ready to Run Programs
by Frederick Holtz
Tab Books Inc., 1983

TI BASIC—Introduction to TI BASIC
by Don Inman, Ramon Zamora, and Bob Albrecht
Hayden Book Company, Inc., 1980

Precollege Computer Literacy: A Personal Computing Approach
by David Moursund
International Council for Computers in Education, 1981

Mindstorms:
Children, Computers, and Powerful Ideas
by Seymour Papert
Basic Books, Inc., 1980

My Friend the Computer
by Jean Rice
T.S. Dennison and Company, Inc., 1976

An Introduction to Computers and Computing
by Jean B. Rogers
International Council for Computers in Education, 1981

The Story of Computers
by Donald D. Spencer
Camelot Publishing Company, 1977

Computer Awareness Book
by Donald D. Spencer
Camelot Publishing Company, 1978

Fun With Microcomputers and BASIC
by Donald D. Spencer
Reston Publishing Company, Inc., 1981

1,2,3 My Computer & Me
by Donna Bearden
Reston Publishing Company, Inc., 1981

Turtle's Sourcebook
by Jim Muller and Donna Bearden
Reston Publishing Company, Inc., 1981

Appendix-C

HELPFUL MAGAZINES

Classroom Computer Learning

The Computing Teacher

Creative Computing

Educational Computer

Electronic Learning

99'er Home Computer Magazine

Appendix-D

SOURCES OF SOFTWARE EVALUATIONS AND EVALUATION FORMS

School Microware Reviews
Dresden Associates

Evaluator's Guide for Microcomputer-Based Instructional Packages
International Council for Computers in Education
University of Oregon
Department of Computer and Information Science
Eugene, OR 97403

Appendix-E

SOFTWARE RECOMMENDATIONS

There are two general classes of software: public domain and commercial. Public domain software is low priced and can be legally copied. The quality varies greatly.

Commercial software is more expensive and usually uncopyable. It is usually more sophisticated and of higher quality than public domain software.

In this section, I first will list some sources of public domain software. I then will recommend commercial software that I have tested.

Sources of Public Domain Software. Often a computer club will have a library of donated and traded software. Check with your computer dealer for the name of a club.

Other organizations offer public domain software for about $10 per disk to cover the cost of the disk and duplication. Once you are a member of the organization, you can buy the public domain disks. Here is one organization that offers public domain software written both in BASIC and LOGO:

Young People's LOGO Association
1208 Hillsdale Drive
Richardson, TX 75081

Recommended Commercial Software. TI produces by far the most educational software for the TI computer. It is also the best designed for the computer in terms of content, structure, and graphics use. Some of the cartridges we are familiar with and would recommend are:

Number Magic (ages 6 to 11)

Touch Typing Tutor (age 6 and up)

You may reach Texas Instruments by calling toll free 1-800-858-4075, from 8 a.m. to 5 p.m. CST, Monday through Friday.

Micro-Ed! also publishes several quality educational programs for the TI 99/4A. Their address is:

Micro-Ed, Inc.
P.O. Box 24156
Minneapolis, MN 55424

Here is another program that we have used.

Crime and Punishment (age 12 and up)
Decision-Making Systems Ltd.
P.O. Box 9557
Wilmington, DE 19809

Appendix-F

GLOSSARY

BASIC The computer language used on most microcomputers

Byte Amount of memory needed to store one character

CAI Computer-assisted instruction

Chip Electronic component made of plastic and silicon

Computer literacy Knowledge of and the ability to use a computer

Documentation Written material that explains a computer program

Drill and practice Software that gives you practice on a skill that you are familiar with

GIGO Garbage in, garbage out

Graphics Pictures on the computer screen

Hard copy Printed listing on paper

Hardware The computer itself: screen, disk drive, etc.

I/O Input/output

K 1,024 bytes

LOGO A good introductory programming language

Machine language What the computer understands

Memory Where instructions are stored

Microprocessor Main part of the computer that carries out instructions

Productivity software Software designed to make you more creative or productive

Public domain software Software that can be copied legally

RAM Memory that can be altered

ROM Preprogrammed memory that cannot be altered

Simulation Software that recreates an event

Software Programs used with a computer

Top down design A method to write computer programs starting with the general idea first and adding details later

Tutorial Software designed to teach something new

Variable Location in memory that holds a value

Word processor A computer with word processing software

Word processing A process of typing and editing text at a computer

INDEX

A

Age level expectations, 16-23
Alpha lock, 4, 14, 15
Arrays, 71-74
Arrow keys, 43, 103
ASCII, 65
Assembler, 37
Auto Plurals, 104-107

B

BASIC, 11, 37, 39, 40
Binary numbers, 11
BK, 80
Byte, 11

C

CALL CLEAR, 47
CALL COLOR, 63-65
CALL HCHAR, 65
CALL HVCHAR, 65
CALL KEY, 77
CALL SCREEN, 63
CAP LOCK, 104
Carriage Return, 146
Chips, 11
COBOL, 37
Colors, 63, 64
Comma, 49
Command mode, 146
Commands, 47
Concatenation, 54
Console, 3
CS, 79

D

DATA, 66-68
Decision, 96
DIM, 73
DISK MANAGER module, 159

E

Edit mode, 146
Enter, 4
Equipment required, 3
External documentation, 98

F

FCTN, 43
FD, 80
FOR NEXT, 62
FORTRAN, 37

G

GOSUB, 77
GOTO, 58, 59

H

Hardware, 9
Hexadecimal numbers, 37
HOME, 84

I

INPUT, 57, 58
Input devices, 10
INT, 70
Integrated circuits, 11
I/O devices, 10

K

Kilobyte, 11

L

LET, 54
LIST, 47
LOGO, 37-39
LOGO editor, 86
LOGRAPH, 118-120
LOGRAPH + WOW, 120, 121
Loop, 97-98
LT, 80

M

Machine language, 11, 36, 37
Make and Take, 107-111
Mass Storage, 3
Memory, 11
Microprocessor, 11
Monster Math, 111-118
MY DEAR AUNT SALLY, 52

N

NEW, 47

O

ON GOTO, 70
Output, 10

P

Parents, 5
Pascal, 37
Peripheral, 11
Peripheral Expansion System, 3
PD, 80
PILOT, 37
Pixels, 113
Precedence, 51
PRINT, 46
Printer, 4

Procedures, 85-88
Processing, 10
Programming, importance of, 35
Programming languages, 36-40
Programming, choice of, 38-40
Programming phase, 98-100
PU, 80

R

RAM, 11
Recursion, 88
REM, 99
REPEAT, 82
RND, 68-70
ROM, 11
RT, 80
RUN, 47

S

Scientific notation, 52
Semicolon, 49
Sequence, 96
Sketch, 126-129
Software:
  advice, 33, 34
  characteristics, 29, 30
  copy protection, 33
  drill, 27, 28
  frustrations, 32, 33
  games, 28
  loading cassette, 14
  loading disk, 15
  loading module, 14
  productivity, 28, 29
  public domain, 34
  simulation, 28
  tutorial, 27
Space bar, 43
Spell Practice, 121-125
Sprites, 82-85
STEP, 62, 63
Steps in Using the Computer, 13
String Variable, 54
Students, 7

T

TAB, 49
TABS, 144
Tape recorder, 10
Teachers, 6
TELL TURTLE, 80
TEXT EDITOR, 157
TEXT FORMATTER, 157
Things to encourage, 23
Things to discourage, 23, 24
TI LOGO, 79
TI WRITER, 5, 136
Top down design, 95-98
Typing software, 134-135

V

Variables, 52-54
Variable Table, 99
Video display, 3

W

Word Processor:
  advantages, 139-141
  definition, 136, 137
  disadvantages, 140, 141
  learning, 141-142
Worksheets:
  BASIC, 46-77
  LOGO, 79-90
  Word processing, 145-157

$12.95

Richard Mowe and Ron Mummaw

ACADEMIC TI®

Take advantage of your TI-99/4A®'s educational potential—at home! *The Academic TI* is an excellent guide written for parents and teachers who are interested in helping youngsters learn with the aid of a TI-99/4A computer. Technical jargon is not used in this book; and the writing style is kept simple. *The Academic TI* discusses a child's growth and development (starting at age 5) and how it relates to the child's ability to use the computer. Information on how to write simple and fun programs is included along with many ready-to-use sample programs—both BASIC and LOGO programming languages are presented. There is a useful section that makes comparisons and recommends educational and commercial software. This guide also discusses typing and writing skills and shows how word processing can greatly improve your child's writing ability. Let *The Academic TI* be your child's guide to becoming a better student!

Table of Contents:

INTRODUCTION
GETTING STARTED
USING COMMERCIAL SOFTWARE
DOING PROGRAMMING
PROGRAMMING IN BASIC
PROGRAMMING IN LOGO
WRITING SOFTWARE
LEARNING TO TYPE
USING A WORD PROCESSOR
SUMMARY

The programs in this book are also available on disk. Please contact the publisher for ordering information.

Illustration by Donna Ward

A Reston Computer Group Book
RESTON PUBLISHING COMPANY, INC.
A Prentice-Hall Company
Reston, Virginia

0-8359-0020-7