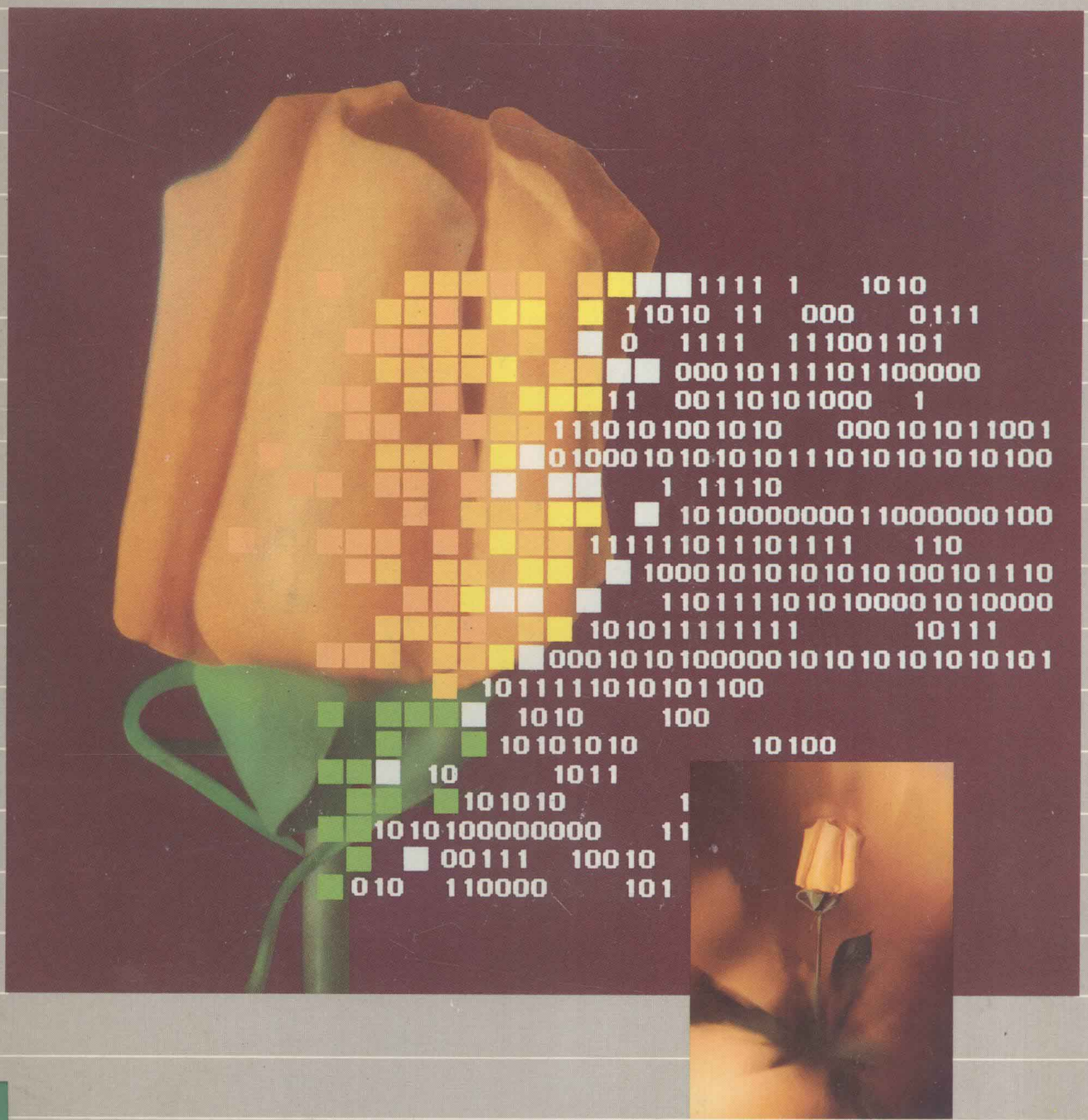
# A Beginner's Guide to WordStar,® 1-2-3,® and dBASE®

## For Computers Using PC-DOS or MS-DOS



Charles S. Parker

# A Beginner's Guide to WordStar,® 1-2-3,® and dBASE®

**For Computers Using PC-DOS or MS-DOS**


## Charles S. Parker

The College of Santa Fe
Santa Fe, New Mexico

The Colorado College
Colorado Springs, Colorado



**Holt, Rinehart and Winston**

New York Chicago San Francisco Philadelphia
Montreal Toronto London Sydney Tokyo
Mexico City Rio de Janeiro Madrid

Address correspondence to:
383 Madison Avenue, New York, NY 10017

ISBN 0-03-013018-2

Printed in the United States of America

Published simultaneously in Canada
7 8 9 0 066 9 8 7 6 5 4 3

CBS COLLEGE PUBLISHING
Holt, Rinehart and Winston
The Dryden Press
Saunders College Publishing

# Preface

One of the most welcome changes in computing during the past several years has been the arrival of *productivity software packages* such as word processors, spreadsheets, and database management systems. These software products have gotten their names because they can increase the on-the-job productivity of nearly anyone who has access to a computer system. Not too long ago, it virtually took a degree in some computer-related field to be able to work directly with computers and the software that controlled them. Today, through the availability of inexpensive desktop microcomputer systems and easy-to-use productivity software, any business person—from clerical worker to company president—can effectively tap the power of computers.

*A Beginner's Guide to WordStar, 1-2-3, and dBASE*—which is to be used as a companion text to *Understanding Computers and Data Processing: Today and Tomorrow,* Second Edition—discusses the basics of three widely used productivity software products: WordStar (a word processor), 1-2-3 (a spreadsheet package with several support functions integrated into it), and dBASE II and III (two database management systems). These packages, which each have the largest user bases in their respective markets, are proprietary. That is, they are each the property of some software vendor and can only be legally operated if an appropriate use license has been acquired from the vendor. Although these software products (and many similar ones) are not free, fortunately they are inexpensive enough to be within the budgets of most users who need them in their work. When you think about it, a $300 package that may save thousands of dollars in typing, filing, or budget-preparation costs is a real bargain.

The first two sections of the *Guide* include some background information relating to using your microcomputer system. Before we get started, keep in mind that you'll have to put in a lot of hard work if you want to fully take advantage of the power that productivity software puts at your fingertips. Although you can learn to do things such as word process a simple letter in an hour or so, it often takes several weeks of practice to use a word processor effectively as a tool for preparing polished, typeset-quality manuscripts. And although you may bumble successfully through your first spreadsheet exercises in a morning or afternoon, it will take considerably more time with spreadsheet software to be able to use it in clever ways that can mean thousands of dollars in labor savings on your job.

Users of this *Guide* are assumed to be working on computer systems with either the PC-DOS or MS-DOS operating systems. The computer system should be equipped with at least 256 kilobytes of main memory (less memory is required if only WordStar is used) and two floppy disk drives.

*Charles S. Parker*

# Contents

# Section 1

# Read Me First: An Introduction

Here you are. First day in the computer lab. Grasped in one hand is a WordStar, 1-2-3, or dBASE program disk that you want to learn to use. In the other hand is a blank floppy disk that will eventually contain the work you do when using one of these programs. Squarely in front of you, a computer system that you must contend with. What to do next?

Relax. If you systematically organize a few notions in your mind before you even touch your computer, every action that you subsequently take won't seem quite so mysterious. For example, think for a moment how strange it would be operating a car for the first time if you never saw a car or a road, and you really didn't understand what purpose cars and roads served. If your driving instructor just said, "Turn the key and put your foot on the accelerator," you'd probably be very nervous. So don't subject yourself to the same sort of torture when learning how to use computers. Sometimes it's better to take a few minutes to learn about the "forest" before you become intimidated by the "trees."

This *Guide* is organized in an easy-to-follow fashion. The discussion of each package begins with what the package is designed to do. Following this is an overview of how the package works—a discussion of the "big picture" (so you can distinguish the forest from the trees later on when you wade through the operational details of the package). Next is a simple, graduated set of examples that will instruct you on the package fundamentals—what you should know about things such as invoking the package from your computer's operating system, executing a few fundamental operations, saving your work, and returning to the operating system. Finally, you'll learn a variety of other useful commands.

Whatever your current level of computer expertise, you should read over Sections 1 and 2 of the *Guide* before you start. If you know very little about computers, read the next subsection especially carefully.

## SOME THINGS YOU SHOULD KNOW IF YOU'RE RELATIVELY UNFAMILIAR WITH COMPUTERS

It's impossible to provide a complete course in computers in just a few pages. So, what we'll do here is touch briefly on some very fundamental concepts—those you must be familiar with just to get started. Whenever

you want to gain a deeper understanding of these concepts, refer to the recommended coverage in *Understanding Computers and Data Processing* where appropriate.

## Hardware, Software, and Files

First of all, let's consider the computer system components you'll be interacting with: hardware, software, and files.

*Hardware* refers to the equipment on which you'll be working. Your computer unit (or system unit), keyboard, monitor, floppy disk unit, and printer are examples of hardware. Shortly we'll be discussing what you should know about each of these pieces of equipment.

*Software* refers to computer programs. The *Guide* discusses three applications software products—WordStar, 1-2-3, and dBASE. We refer to them as *applications software* because they tell the computer how to perform a specific, job-related task (application), such as preparing letters or maintaining employee records. Here, the terms productivity software and applications software will be used interchangeably. It's likely that the applications software packages you'll be working with are encoded on floppy disks. These software disks are commonly called *program disks.*

There is also another type of software—*systems software.* You must interact with systems software to gain access to WordStar, 1-2-3, dBASE, or any other applications software. The piece of systems software that you'll be most concerned with here is your *operating system,* which is named either PC-DOS or MS-DOS. Learning to use your operating system effectively is just as important as learning how to use any applications software package. It enables you to do things such as prepare new disks for use on your computer system, copy the full or partial contents of one disk onto another disk, and determine what is on your disk.

The term *files* refers to the material you'll be creating with applications software. With a word processor the files you create are such things as letters, manuscripts, and book chapters; with spreadsheet software, budgets, asset amortization tables, and profit-and-loss statements; with database management systems, a group of employee records, vendor records, and client records. Generally, you'll be storing these files on the blank floppy disk you purchased or were given. In Section 2, you will learn how to format this disk, thereby making it usable on your computer system.

In a two-disk-drive system most people reserve the "first" (leftmost or topmost) disk drive for the program disk and the "second" (rightmost or bottommost) disk drive for the disk to contain the data files. This convention should be followed rigorously if you wish to avoid clumsy mistakes. The first and second drives are commonly referred to as the *A drive* and *B drive* respectively. Also, many users cover the *write-protection-notch* (see *Understanding Computers and Data Processing,* pages 136 and 138) on their program disks to ensure they don't accidentally destroy the programs on them.

### Primary and Secondary Memory

The computer system you'll be using employs electronic *memory* to store both data and programs. If you are a beginner to computers, you must learn the differences between your system's primary memory and secondary memory.

Physically, *primary memory* (often referred to as *main memory*—or sometimes *RAM,* for *random-access memory*)—pertains to the storage chips that are mounted onto boards inside your computer's system unit, while *secondary memory* usually refers to disk storage. The conceptual difference between the two is that primary memory is temporary and stores only what the computer system is currently working on while its power is up. Secondary memory, on the other hand, is semi-permanent. It doesn't change unless you specifically summon the computer system to "write" (save) something onto it. The computer system itself generally controls the contents of primary memory whereas you control the contents of secondary memory.

When your computer's power is shut off, the contents of primary memory will be lost, but what you have stored on disk won't be erased. Later you'll learn how to save files on disk and how to retrieve them. Both saving and retrieving can be done through either an applications software command or an operating system command. One of the most frustrating experiences for any newcomer to computers is turning off the computer's power (thereby erasing the contents of RAM) before an important file has been saved onto disk. Once you have saved something onto disk, it is an easy matter to load it back into primary memory again.

Primary and secondary memory capacities on your computer system are measured in *bytes.* Roughly speaking, a byte is equivalent to a single character of data. So, for example, a 300-character memo prepared on WordStar occupies roughly 300 bytes of storage space. Often memory capacities are stated in thousands of bytes, or *kilobytes* (KB). Two common capacities are 256 KB RAM in the computer's system unit and 360 KB for floppy disks.

## LEARNING PRODUCTIVITY SOFTWARE: SOME TIPS

Often before you learn anything new, you'll look to others for some general advice on how to proceed. Below are four tips that you may find helpful when learning to use productivity software.

### A Few Commands Go a Long Way

Instead of trying at the outset to learn as many commands as you can possibly pick up, first try mastering the few commands you'll need to do the exercises at hand. You can pick up others later as you encounter problems that necessitate using them.

For example, with WordStar, you'll first be creating small documents, so that your keyboard's arrow keys will be sufficient to move around through the text (a character at a time). However, as your documents get longer, you'll need to move through text faster. This will necessitate learning to use special keys and commands that will send you through large blocks of text at a time. Software designers generally try to make their products easy to use. If you find yourself doing something that seems very labor-intensive, that's when it's time to look around for a more powerful command.

Oddly enough you'll find that mastering just 15% or so of the available commands in any package will probably enable you to do 85% or so of the work you'll ever need or want to do with the package.

## Resources

Say you do want to learn a more powerful command and you can't find it in this *Guide*. Where do you go? Fortunately, there is a rich variety of resources available on WordStar, 1-2-3, and dBASE to enable you to go well beyond what you learn here. The popularity of these packages ensures that a wide assortment of instructional materials will be present in the marketplace. Five useful types of resources are disk tutorials, video cassettes, training manuals, trade books, and reference manuals.

Several disk tutorials are available for each of the productivity software packages covered in this *Guide*. To use one of these instructional disks you put the disk into a disk drive (usually the A drive), invoke the tutorial through a special command, and follow the instructions that will subsequently appear on the screen. Generally the disks tell you which keys to press and explain to you the resulting output from each key sequence. Tutorial disks are available through both productivity software vendors and "third-party" training companies (such as American Training International in Manhattan Beach, California and Cdex Corporation in Los Altos, California).

Both training manuals and instructional video cassettes are like tutorial disks in that they are step-by-step learning tools. As with tutorial disks, many third-party companies produce training manuals and video cassettes for popular productivity software packages. The quality of some of these third-party products can be exceptional, since the companies producing them live or die on how well they are able to train people.

There is a wealth of trade books available on WordStar, 1-2-3, and dBASE. These books can be quite useful because they may go through a very large and comprehensive set of examples to make you familiar with a package. Also, they may cover extensively a particular aspect of a package that you want to learn about in some depth. Since there are so many trade books now available, there is a good chance that you'll find one written at a level that meshes well with your level of understanding. Your best bet in locating such books is to wander down to the computer section of your local bookstore and browse. Also, some college and university libraries stock such books along with their other holdings.

**USING PRODUCTIVITY SOFTWARE: BASIC OPERATIONS**

- *Accessing* the package from the operation system.
- Informing the package that you either want to *create* a new file or *retrieve* an old one from disk.
- Commanding the package to *save* a file onto disk.
- Commanding the package to *print* a file.
- Commanding the package to *delete* a file that's on disk.
- Indicating to the package that you want to *quit* working on your current file and do something on another file.
- *Leaving* the package and returning to the operating system.

***Figure 1-1. Fundamental productivity software operations.***

When all else fails in your search for an answer, you must often resort to the comprehensive reference manual provided by the vendor of your software package. Vendor reference manuals generally are written on a more technical level than trade books, but they are the final word on what a package can and can't do. If the reference manual fails you, call up a knowledgeable friend or the software vendor.

## Experiment Freely

A joke sometimes told about high-level business executives is that they tinker first and (if not successful) read the instructions later. This technique can actually be extremely helpful when learning how to use a productivity software package. These packages often beckon you to do things that require you to use your intuition. Using your intuition will often produce faster results than looking something up and reading several paragraphs. If you're unsure about how to proceed beyond a certain point, and the cost of making a mistake isn't high, try something that seems reasonable. People who wait until they're absolutely certain about how to proceed before taking any action just won't learn these packages very fast.

## Packages Have the Same Fundamental Operations

Finally, you should observe that most productivity software packages have in common seven basic operations. These are shown in Figure 1-1. In the *Guide* these operations will generally be the first ones you learn as you study WordStar, 1-2-3, or dBASE. Depending on the package you use, the file that you work on may be a word-processed *document*, a *spreadsheet*, or a *set of records*.

# Section 2

# The Operating Environment

Before you dive right into working with an applications software package, it helps to know something about the operating environment in which these packages work. Although there are a great number of subjects that we could discuss about this environment, we'll examine only two here: the support hardware available with your microcomputer and the PC-DOS and MS-DOS operating systems.

## SUPPORT HARDWARE

Most likely, the hardware configuration you'll be working on includes the following items:

- A system unit
- A monitor
- A printer
- A keyboard
- Two floppy disk drives (which may be built into your system unit)

These devices are shown in Figure 2-1. We'll now discuss each of them.

### The System Unit

Your microcomputer's *system unit* was described in some depth in Chapter 4 of *Understanding Computers and Data Processing,* pages 111–118. This device is the "box" that contains the system board (with your microprocessor and main memory) as well as a variety of related circuitry.

When you're working with any productivity software package the single most important operational detail you should know about the system unit is the location of the power switch. The system unit (and monitor and printer, as well) has a switch that turns its power on and off. You'll either have to turn on your system unit, monitor, and printer individually or, if your computer system has a *power manager* that controls all of these devices, you can flick one switch to turn them all on or off together. Remember, the system unit contains the computer's primary memory, so when you turn the system unit off, the contents of this memory are erased.

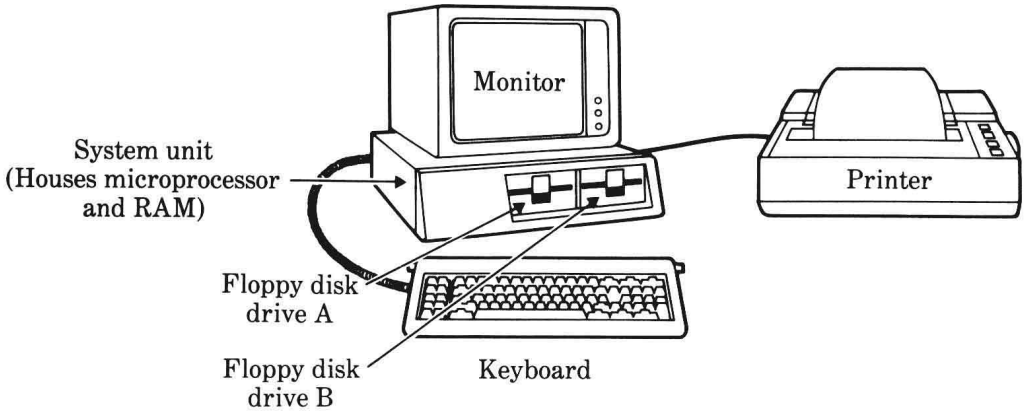



***Figure 2-1. A typical microcomputer system.***

## The Monitor

*Monitors* resemble TV sets (see Chapter 6 of *Understanding Computers and Data Processing*). Besides having a power switch, most monitors have contrast and brightness knobs that enable you to adjust the image of the screen to suit your visual preferences. Monitor screens are either *monochrome* (one foreground color) or *full-color* (eight or more colors).

## The Printer

In recent years *printers* available for microcomputer systems have improved dramatically. There are so many variations among printers (see Chapter 6 in *Understanding Computers and Data Processing*) that it is impossible to catalog all of them here.

Most likely you will be working on a low-speed, *impact dot-matrix printer.* Many of these have features to provide an assortment of print typefaces—including both draft-quality and near-letter-quality (NLQ) print, boldfacing, subscripts and superscripts, compressed characters, elongated characters, graphical characters, and so on. If you are interested in exploring the features of your printer further, consult the operations manual available from its vendor.

## The Keyboard

The *keyboard* of your microcomputer system is an extremely powerful device. Most keyboards are controlled by specialized computer chips. Chances are the keyboard you'll be working with looks something like the one in Figure 2-2. This is the keyboard available with the IBM PC. On many keyboards the keys are *typematic;* that is, they repeat as long as you hold them down.

Many keyboards have three sections. In the center is the *typewriter section.* The arrangement of the keys is similar to that on a conventional typewriter. There are, however, many special keys that you may not have

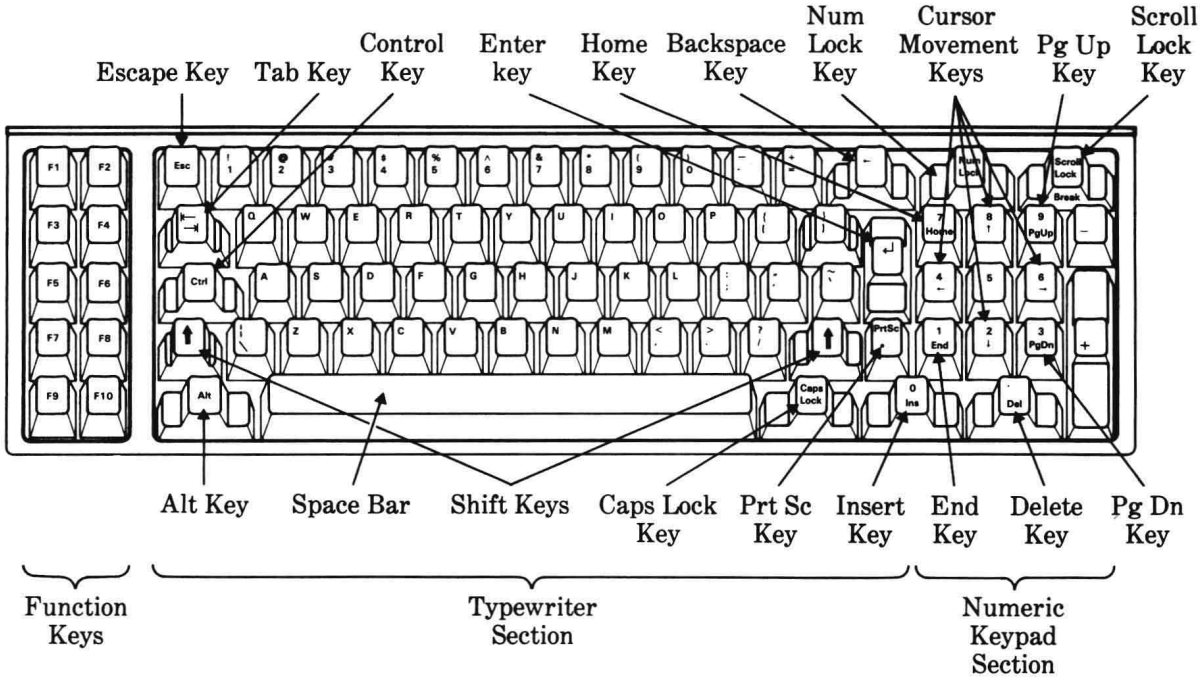



***Figure 2-2. The IBM PC keyboard.***

encountered before, such as the Alt, Esc, and PrtSc keys. We'll discuss these and other special keys shortly. One thing you should know right now, however, is that every productivity software package uses these special keys in its own way. For example, the Del key works completely differently in WordStar than it does in dBASE.

On the keyboard that accompanies the IBM PC, to the left of the typewriter section are the *function keys*. These, too, are special keys and each of them has a different meaning in each software package. For example, if you are using WordStar on the IBM PC, the F2 key is used to block-indent paragraphs. In 1-2-3, the F2 key is used to switch into the edit mode.

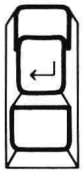
Not all keyboards have 10 function keys as does the IBM PC. Also, these keys may be positioned differently than they are on the IBM PC's keyboard and may even have different meanings for the same software package. For example, the keyboard available with the TI Professional Computer has 12 function keys, located at the top of the typewriter section rather than to the left of it. What's more, if you use WordStar on your "TI Pro," the F2 key will reformat paragraphs, not block-indent them as on the IBM PC.

In spite of these differences, function keys are really easy to use and tremendously helpful. Each of them is designed to perform an operation on a single keystroke, saving you keying time. In many applications software packages, the operations performed by the function keys are shown at the bottom of your monitor screen.

The *numeric keypad section* is a cluster of number keys and special keys. On most keyboards it is located to the right of the typewriter section. The numeric keypad is handy to use if you are trained—as are many people—to enter numbers from a cluster of keys arranged in this configuration.

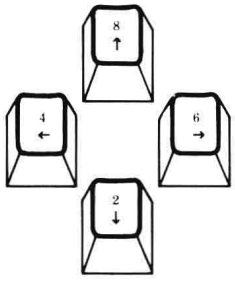
As we discuss each software package we'll be mentioning the significance of several of these special keys. There are, however, a number of important keys and keystroke sequences that you should be familiar with right away. These are:

Enter Key



The **Enter** (or **Return**) key is used to enter commands to the computer system in whatever software package you'll be using. In many cases, before this key is depressed, everything you type is held in a special memory area called an *input buffer,* awaiting corrections, before being sent on.

Arrow Keys

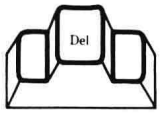



The four **arrow keys** are used to move some pointer—such as a cursor or highlight—around the screen. In 1-2-3 these keys can also be used effectively at certain times to enter data.

Escape Key

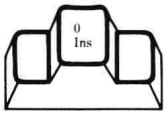

The **Escape** key is commonly used to abort some action that you've previously taken at the keyboard. In 1-2-3 you can also use this key to backtrack to some previous point in a command sequence.

Delete Key



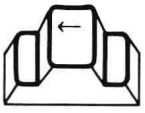

The **Delete** key enables you to remove characters. In 1-2-3 and dBASE this key removes characters at the cursor position. In WordStar it removes characters to the left of the cursor position.

Insert Key



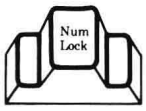

The **Insert** key allows you to insert characters. In WordStar and dBASE, this key enables you to insert characters at the cursor position. Also, it works as a *toggle key,* that is, it activates a command (insertion) when you depress it once and deactivates the command when you depress it again. In 1-2-3 the Insert key is inactive.

Backspace Key



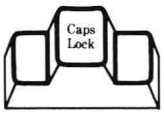

The **Backspace** key moves the cursor back one space. In 1-2-3 and dBASE, this key will also erase characters as it backspaces to their positions. In WordStar this key erases characters when you are naming files but not when you are editing documents.

NumLock Key



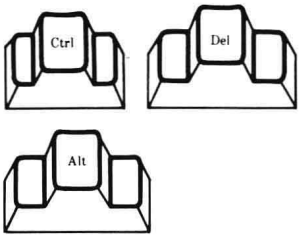

The **NumLock** key is used to activate the numeric keypad section of the keyboard. This key also can be an annoyance because it can easily be hit by mistake and deactivate the arrow keys. So, if depressing the arrow keys produces numbers on your screen, hit the NumLock key again (it's also a toggle key) to deactivate the numeric keypad.

CapsLock Key




The **CapsLock** key is used to produce uppercase letters on your keyboard, saving you from having to depress the shift key. This key can be very handy when typing titles that consist only of uppercase letters. CapsLock is also a toggle key, so you can deactivate it by hitting it again.

Ctrl-Alt-Delete
Key Sequence




If you hold down the **Ctrl** and **Alt** keys (which work like shift keys) and then hit the **Delete** key, the sequence will cause a *warm boot* of your computer system. This is virtually equivalent to turning your computer off and then on again. Probably everyone working on a microcomputer system has faced some situation where proceeding with an applications software package was not possible, necessitating this drastic action. If you execute this key sequence, everything you currently have saved in main memory will be lost.

Shift-PrtSc
Key Sequence


The **Shift-PrtSc** sequence of keys enables you to send whatever is currently on your monitor screen to the printer. If what you want to send to the printer can fit onto a single screen, and you're not overly concerned about getting some extraneous characters that you see on the screen on your output, this is the best (and the fastest) way to get output. Using this sequence of keys provides an excellent way to copy a file or error message from the display screen, so that you can later discuss it with your instructor.

### The Disk Drives

Many *floppy disk drives,* such as the ones in Figure 2-1, are built into their system units. Many system units contain two disk drives and, as already mentioned, these are referred to as the A and B drives. The computer system can only be "pointing" to one of these drives at any time for reading or writing. The drive that the computer system is currently pointing to is called the *logged drive* or *default drive.* The next subsections, on DOS, discuss the disk operations and file-naming procedures you must know when working on your computer system.

Inserting a disk into a floppy disk unit and caring for *floppy disks* was covered in Chapter 5 of *Understanding Computers and Data Processing,* pages 138–141. If you've never worked on a floppy disk system before, you should carefully read these text pages. Four things you absolutely need to know are:

- The disk unit will only work if the disk is installed properly in the drive.
- You should never attempt to remove or install a disk in a drive when that drive's *indicator light* is on.
- Never touch the surface of the disk that is exposed in the disk's *recording window.* Like the phonograph records or musical tapes you have at home, floppy disks should be treated with respect.
- You cannot write to your 5¼-inch floppy disk if the disk's *write-protect notch* is covered.

## THE PC-DOS AND MS-DOS OPERATING SYSTEMS

The operating system is your gateway to all of the applications programs on your computer system. It also manages your disk files, enabling you to save, retrieve, copy, and erase them. You work with your operating system—either PC-DOS or MS-DOS—by issuing DOS commands (see the next subsection). *DOS,* incidentally, stands for *D*isk *O*perating *S*ystem.

Both PC-DOS and MS-DOS were developed by Microsoft Corporation, of Bellevue, Washington. Except for a few minor differences, these operating systems are virtually identical. PC-DOS was created originally for the IBM PC, while MS-DOS was devised to operate on computers that look and work a lot like the IBM PC. A few examples of such computers are the TI Professional Computer, Leading Edge PC, Compaq Deskpro (Model 2), Compaq Portable, AT&T 6300 PC, Tandy 1000, and Zenith Z-150.

Both the PC-DOS and MS-DOS operating systems have been revised many times since they first were developed. Each major revision is referred to as a *version* (sometimes also called a *revision* or *release*). All versions starting with the number 1 (such as 1.0 and 1.1) were designed for the

earliest 16-bit microcomputer systems, which used only floppy disks and cassette tapes for secondary storage. In late 1983, when IBM came out with its PC XT computer, a machine that featured a built-in hard disk, version 2.0 of DOS was developed to deal with the intricacies associated with storing data on a large-capacity secondary storage device. Improved revisions of this "hard disk version" of DOS (such as 2.1, 2.2, and 2.3) followed. Later, in 1984, when IBM announced its multiuser IBM PC AT computer, an entirely new set of DOS versions (such as 3.0 and 3.1) followed.

All of these versions of DOS are "upward compatible." What this means is that later versions of DOS can work with both applications programs and data files that were created on earlier versions. Whichever version of DOS you use, it will probably be contained on a floppy disk, which is commonly referred to as the "DOS disk" or "system disk."

To start your computer system and work with DOS, place your DOS disk in the A drive of your computer's system unit, close the drive door, and turn on your system unit (as well as your monitor). You'll now hear some noise taking place inside the system unit before the first message is displayed on your monitor screen. What is happening here is that you are *booting up* your system. What takes place during the booting-up process is that certain operating system commands are read into main memory, so that the computer can access them quickly.

The commands that the operating system boots into its main memory are called *internal commands.* You can issue these commands at any time your computer system is on, no matter what disks are present in the drives. The commands that are not stored in internal memory when the system is booted up are called *external commands.* For you to issue these commands, the DOS disk must be present in one of the drives. By convention the A drive is used, since DOS, too, is a program.

Once your system is booted up, you'll see a message indicating the version of DOS you are using, followed by a prompt asking you to enter the date. If, for example, the date is February 1, 1987 you would enter this date as follows:

**2-1-87 (cr)**

where (cr)—for carriage return—represents depressing the Enter key.

DOS is particular about how you enter the date, and you have to enter it in the form shown. When you enter the date in this way, DOS will record it on your disk's file directory when you save files. If you don't wish to enter a date, hit the Enter key, and DOS will proceed to prompt you for the time.

The time is entered in a similar manner. Time is expressed in hours:minutes:seconds.hundredths. Hours can be between 0 and 23; minutes and seconds between 0 and 59. Most people omit the hundredths entry and, often, the seconds entry. So, for example, if it is 10:35 A.M. you enter the time as

**10:35 (cr)**

Or, if it is 6:23 P.M., you would enter

**18:23 (cr)**

If you don't wish to enter a time, hit the Enter key to proceed.