Microcomputer Power

$16.95

Easy to follow, step-by-step instructions
Use the VisiCalc worksheets to

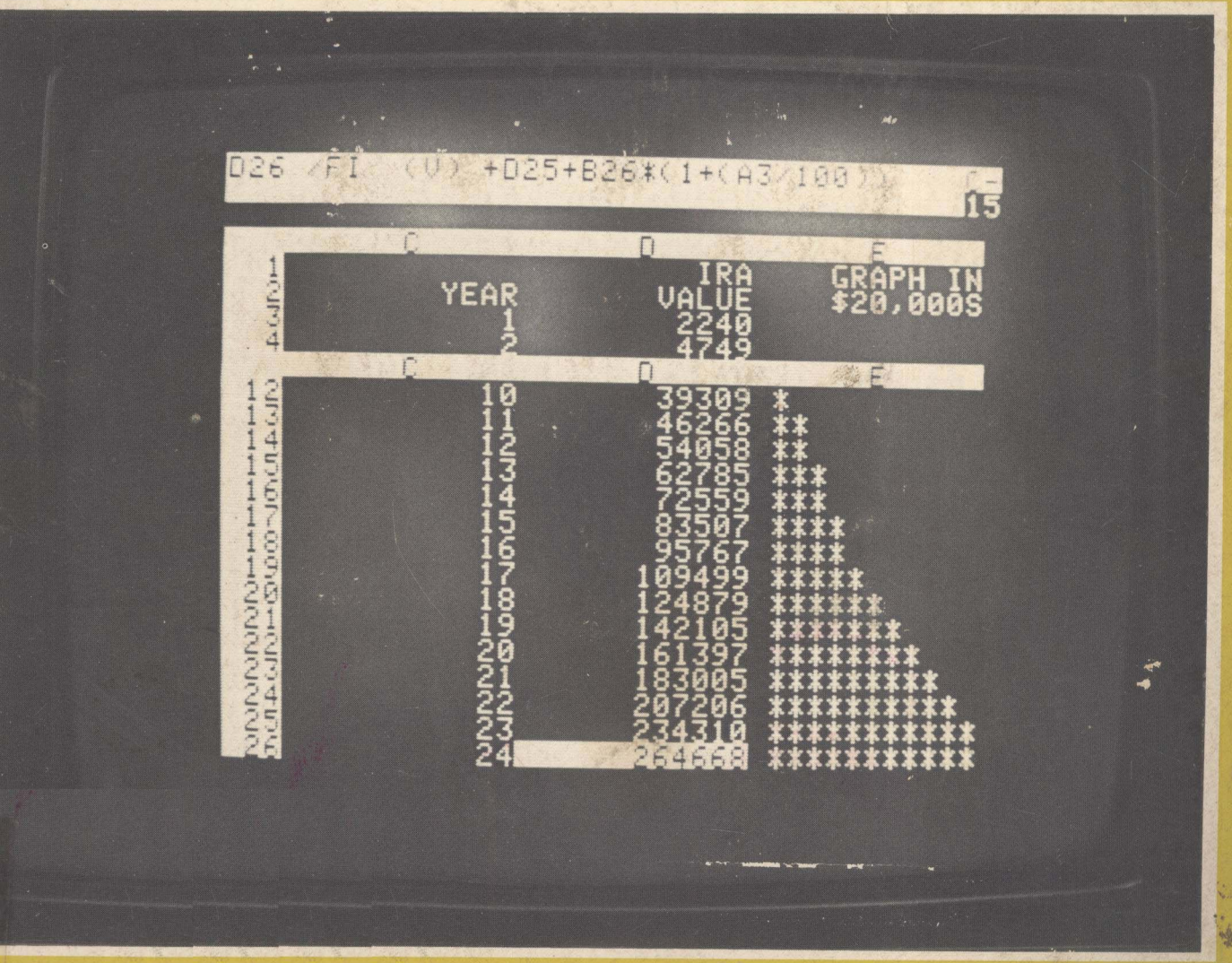
- calculate sales projections
- evaluate your portfolio
- log your expenses
- compute your IRA account
- determine your net worth

Written for the novice and the professional.

VisiCalc®

for the Apple® II Plus Computer

Edouard J. Desautels



Microcomputer Power

VisiCalc®
for the Apple® II Plus Computer

Edouard J. Desautels
University of Wisconsin
Madison

wcb
Microcomputer Power Series
Wm. C. Brown Company Publishers
Dubuque, Iowa

Microcomputer Power Series

Second Printing, 1983

Library of Congress Catalog Card Number: 82–72690

ISBN 0–697–09968–7

Printed in the United States of America

2-09968-02

Preface

The VisiCalc® program makes a computer as easy to use as a calculator, and it gives you far more power than a calculator does. This little book is intended to show you how you can use the VisiCalc program effectively. This book is designed so you do not have to have any previous experience with computers to use the VisiCalc program.

The first chapter sketches the kind of situation in which the VisiCalc program excels. The next chapter shows you how to start using the Apple II Plus Computer. Then we start looking at the specifics of using the VisiCalc program. Although it is easier to keep reading the chapters in the order in which they appear, you can often skip ahead if you wish to look into some feature of special interest to you. A comprehensive index will help you find your way.

Most of the features of the VisiCalc program are presented while solving a sequence of realistic problems, such as evaluating the accumulation in an IRA (Individual Retirement Account). Some of these problems are solved a second time, so you can better appreciate the contrast between different approaches.

Almost everyone has to come to grips with numbers, lots of them. The VisiCalc program is such a powerful assistant in helping you manage numeric information that you should seriously consider equipping yourself with this tool. Perhaps this book will help you make such a decision.

Although this book contains detailed instructions on how you can yourself do everything that is shown, an optional diskette may be used to reduce the typing which would otherwise be required. See the appendix for further information. It is assumed that you have already purchased the appropriate version of the VisiCalc software. If not, you may obtain it from your Apple II Computer supplier.

Contents

Chapter 1

WHY USE THE VISICALC PROGRAM?

Electronic computers have been in use for over 30 years. Why is it that the VisiCalc program is one of the best-selling computer software packages since the beginning of the computer age? The Time magazine article on the VisiCalc program gives you some idea of its financial success. What makes the VisiCalc program so attractive?

The Smash Hit of Software

Daniel Bricklin, 29, and Robert Frankston, 31, a team of new-wave composers, have penned a dynamite disc that has grossed an estimated $8 million. It is not a punk-rock smash, but an unmelodic magnetic number called VisiCalc, the bestselling microcomputer program for business uses. The featherweight sliver of plastic is about the size of a greeting card, but when it is placed in a computer, the machine comes alive. A computer without a program, or "software," is like a $3,000 stereo set without any records or tapes.

Three years ago, Bricklin, then a first-year Harvard Business School student, conceived VisiCalc while struggling with financial-planning problems on his calculator. He enlisted the aid of Frankston, a longtime friend and an expert programmer, to develop a new piece of computer software that would make juggling all those figures easier.

The partnership paid off. Since late 1979 nearly 100,000 copies of nine different versions of VisiCalc have been ordered at prices ranging from $100 to $300. It is far ahead of other business programs like Data Factory and General Ledger, and even outsells the programs for Star Cruiser, Dogfight and other arcade-like computer games.

VisiCalc translates simple commands typed on a keyboard into computer language that the machine then uses to solve problems. It enables a businessman, for example, to manipulate labyrinthine equations to calculate financial trends for his company. If he changes one figure, the machine can tell quickly how that affects the other numbers. A firm that gives its workers a 10% pay hike could estimate how that action would alter its costs, sales, profits, or dividends.

The computer program is being put to a wide range of uses. It helps Allerton Cushman Jr., a New York financial analyst, to project insurance-industry profits during the week and tote up his income taxes on the weekend. The Cabot Street Cinema Theatre in Beverly, Mass., bought VisiCalc to figure out which pattern of movie show times draws the best box-office receipts. An accounting firm in Las Vegas plans to use VisiCalc to tell its gambling-house clients how to position slot machines around the floor to ensure the biggest take. VisiCalc is obviously one composition that is in no danger of fading from the charts.

Reprinted from the March 2, 1981 issue with permission of Time, Inc.

It is rarely the case that financial success is also accompanied by formal recognition of excellence by one's academic peers. The impact of the VisiCalc program is such that the ACM, the Association for Computing Machinery, the foremost professional society of computer scientists, awarded its 1981 Grace Murray Hopper award to Daniel S. Bricklin, the chairman of the board of Software Arts, Inc., which originated the VisiCalc program. He received this award for the excellence and elegance represented by

the development of the VisiCalc program. It is worth noting that this is the first time that any activity involving microcomputers or personal computers has merited the ACM's attention. It is almost as if very small computers were previously regarded as toys. This is definitely no longer the case.

What is it about the VisiCalc program that sets it apart from ordinary computer programming languages? There can be no doubt that any solution that the VisiCalc program can produce can also be produced by writing a computer program to generate that solution. For any problem you might wish to solve using the VisiCalc program, a computer program to solve that problem could be written, using some computer programming language such as BASIC, COBOL, FORTRAN, PASCAL, etc.

That is exactly the reason for using the VisiCalc program. You **don't** have to write a program to solve your problem when you use the VisiCalc program. You key in the relevant data (there is no way to avoid this completely, no matter what you are using). You then specify how the data items are related, and what answers you want calculated. At this point, the computer expert might object: "This is the same as writing a program." Theoretically that may be true; in practice, it makes all the difference in the world. Writing computer programs can be very tricky and time-consuming, and that is after you have invested your time and effort in first learning the programming language.

When you want to solve a new problem using a computer, you usually have to write a program to solve that problem, using one of the programming languages we mentioned earlier. That usually means you also have to use some kind of editing program just to type in the program you need. Then you may have to prepare a data file, with the help of the editing program.

The data file is considered input to the program you wrote, and the results produced by your program are called its output. We often depict the input as flowing into the program which has been placed into the computer's memory, and show the results produced by your program as flowing out from it, as we see in figure 1.1.

Figure 1.1 Input, program, and output relationships

After you examine the output from your program, you may come to one of several conclusions:

(1) great- let's stop computing.
(2) oops!- there must be a "bug" in the program; try to find it and fix it.
(3) it's ok, but what if ...?

With the VisiCalc program, you are more likely to get the first conclusion first. Number two is much less probable, simply because the VisiCalc program won't let you request many ridiculous computations. The VisiCalc program really shines in the third situation. You simply change the desired number or formula, and you immediately see the consequences. You don't have to fool with an editing program, or even request that your program be rerun.

Case 3 might involve wondering what would happen if some data item had a slightly larger value, or what would happen if the formula in the program was just a little different.

With the conventional approach, the input data is laid out for the convenience of the computer program (or whoever wrote the program). The actual layout of the input data has little if any spatial relationship to the results. With the VisiCalc program, you begin by putting the numbers where they should be at all times. And you decide where the results should appear, in relationship to the other input. If you should change your mind, it is a simple matter to move things around painlessly. If you then decide to change a number, you simply locate its old value where you expect to find it (not at some strange location chosen for a program's convenience). As soon as you change that number, all the other outputs which depend on that number immediately change.

Perhaps the simplest yet the most accurate way of describing how you use the VisiCalc program is to think of the computer keyboard as a pencil, with one of the keys used as an eraser, and the screen is a blank sheet of paper that seems to be as wide and as long as you need. If you forget something, it is easy to cut the sheet and paste in new material. It is not hard to move things around. When you ask for some result to be calculated, your electronic worksheet will remember how the calculation was done. So if you later change one of the values that the calculated result depends upon, the computer will recalculate the new result with no further ado. It actually makes working with numbers pleasant!

A Brief Example

The following simple example will give you a better idea of the difference between solving a problem using the VisiCalc program versus solving a problem using the conventional computer programming language approach.

Suppose you were a budding author, and had just had your manuscript accepted by a publisher. The publisher might have proposed that you be paid royalties based upon the following sliding schedule:

Earn 5% of selling price for the first 3,000 copies.
Earn 7% " " next 4,000 "
Earn 8% " " next 5,000 "
Earn 10% " " for all additional copies.

With the VisiCalc program, you would lay out the essential data in the form of a table, as we see here in table 1.1.

Rate	Copies
5	3,000
7	4,000
8	5,000
10	?

Table 1.1 Raw data

You would then make an educated guess as to what number to use for the ? in table 1.1. You might like to project your potential income (before taxes) if the book sold say 50,000 copies, assuming that it sells for $1 per copy (it is a very small book). So the ? is replaced by 50,000-(3,000+4,000+5,000), giving us 38,000. You then ask the VisiCalc program to work out the product (Rate multiplied by Copies) for each line, which results in table 1.2, since our rate figures are actually percentages.

Rate	Copies	Income
5	3,000	150
7	4,000	280
8	5,000	400
10	38,000	3,800

Table 1.2 Projected income

Of course, what you really want is a running total, so you ask the VisiCalc program to produce another column headed "Total" which is to show for each row the sum of the current and all preceding income figures. That being done, you now see the results in table 1.3.

Rate	Copies	Income	Total
5	3,000	150	150
7	4,000	280	430
8	5,000	400	830
10	38,000	3,800	4,630

Table 1.3 Projected total income

This may seem like much ado about nothing, at this point. You could have done the same thing with a calculator, or merely with paper and pencil. True--but having gone this far, you can now begin the "What If ..." phase. As the budding author, you might want to negotiate a better royalty schedule.

So you begin thinking "What if my first book is not a terrific success? I should ask for a higher percentage for the first few thousand sold, just to be safe." Suppose you settled on 6, 8 and 9%, in place of the 5, 7 and 8 that was offered; you don't want to be greedy. You can now proceed to replace the 5, 7, and 8 by the 6, 8 and 9, and, lo and behold, you immediately see the consequences, as shown in table 1.4.

Rate	Copies	Income	Total
6	3,000	180	180
8	4,000	320	500
9	5,000	450	950
10	38,000	3,800	4,750

Table 1.4 Projected total using new rates

You might think about these totals for a while, and because you are now convinced you have a best seller on your hands, perhaps it would be better to focus on negotiating a better top rate than the 10% that was offered. After all, the other rates will make very little difference if you sell 100,000 copies. So you change the 10% to 20% (dreamer) and you decide to increase the 38,000 by 50,000. The results are shown in table 1.5.

Rate	Copies	Income	Total
6	3,000	180	180
8	4,000	320	500
9	5,000	450	950
20	88,000	17,600	18,550

Table 1.5 New projected totals

If you were in the publisher's shoes, you would also like to be able to do the same juggling of figures. As the budding author, you have much less experience with the consequences of manipulating either the rates, the sales thresholds, or the number of steps in the schedule. The publishers can probably do it in their heads; you could use a little help.

You could of course have written a computer program to perform these calculations. For many computer programs, you have to lay out the numbers to be processed in a data file. The data file for this program could very well look like the following list (using the first set of rates and copies):

```
5,3000,7,4000,8,5000,10,38000
```

This is awkward to read and change. It might have looked like:

```
05 3000      or        05070810
07 4000                03000 04000 05000 38000
08 5000
1038000
```

where you find yourself putting in leading zeroes or leading

blanks with the data, because the exact spacing between these numbers may be critical, depending upon the programming language that was used in writing the program. You could easily provide data which looks correct to the naked eye, but which the program takes to be ten times larger (or ten times smaller) than what you had in mind.

The data file

```
5                    3000
7                   4000
8                    5000
10                  38000
```

seems to have the right numbers,even though the alignment for the 4000 and the 10 is a little sloppy. Some computer programs would interpret this 4000 as if you had written 40,000 because the actual position of the number on the line was critical. This is much less of a problem with the VisiCalc program. You immediately see what the VisiCalc program thinks of the number you just typed. If it is not what you intended, you can change it right away.

What About Big Problems?

Suppose you had a really big problem to solve? Could the VisiCalc program handle it? Suppose you were going to be a real-estate tycoon and you were working out a ten-year projected statement of cash flow. Such a statement is a table with at least ten columns of numeric data, and many dozens of rows, depending upon how much detail you want to include in projecting your cash disbursements. Then we also need some labels to keep track of things. If each column is to hold numbers as large as eight digits, and if we leave a little space between columns for ease of reading, the cash flow table will be about 120 characters wide. Since most computer video display terminals (VDTs) can only display 24 or 25 rows of 80 characters at one time, you would be hard pressed to squeeze all of the cash flow table onto the VDT screen at one time. Imagine how much harder it might be with an Apple II with a 40-column display! As it turns out, you can do amazing things with an Apple II equipped with the VisiCalc program.

With the VisiCalc program, you can construct and display tables with as many as 63 columns and 254 rows and you can "browse" over parts of the table very easily. If the whole table won't fit on your screen, the VisiCalc program treats your VDT screen as if it were a "window." Your screen window lets you see any part of the table you wish to see, getting as much of it as will fit on your screen at one time. Trying to do this with a conventional computer program would be far more difficult. Figure 1.2 illustrates the idea of a window.

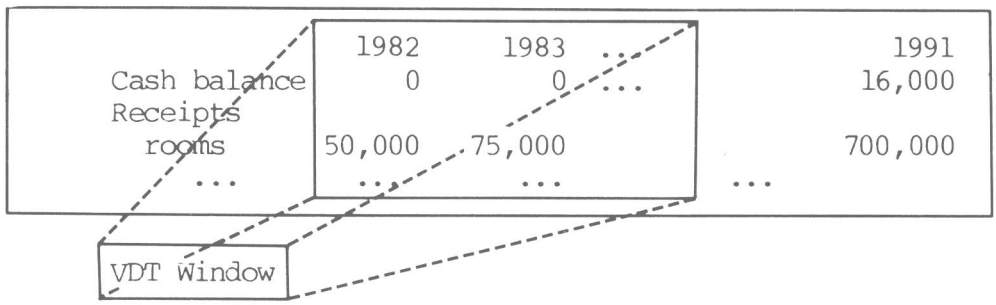


Figure 1.2 VDT Window into a large table

Coming Up

Specifics on using the VisiCalc program: how do you tell the VisiCalc program to do something? What if you made a mistake? What if you were in the middle of filling out a worksheet and had to leave suddenly: how can you save worksheets, and recall them later? These and many other features of the VisiCalc program will be described, discussed and illustrated as we proceed. It goes without saying that you will learn more, faster and better, if you can be using the VisiCalc program and trying the things we are discussing. We hope however to provide sufficiently detailed examples so that you can follow what is going on even if you don't have immediate access to a computer equipped with the VisiCalc program.

SUMMARY

The VisiCalc program is data-oriented; using it is very much like using a calculator. You begin with your own raw data, lay it out on what amounts to an electronic worksheet, using your VDT's screen as an easily erasable worksheet. Then you begin specifying the relationships between your data and the desired results. You build up to the desired end-product in a step-by-step fashion, seeing the results at every step. You always see your input data in the natural spatial relationship it is intended to have with respect to any computed results.

Problem solving with computers, using the conventional approach, is program-oriented rather than data-oriented. Most people who are not computer experts feel more at ease with the data they know well than with the use of unfamiliar computer programming languages. Most people are familiar with the everyday use of a simple worksheet. The VisiCalc program combines the ease of using a calculator and the familiarity of a worksheet with the power of a computer. It follows that most people will find that the VisiCalc program provides a very natural, user-friendly way to make the computer work for you.

Chapter 2

GETTING ACQUAINTED WITH THE APPLE II COMPUTER

This chapter deals with the operation of the Apple II or Apple II Plus Computer and it introduces some new ideas. No previous experience with computers is required to use the VisiCalc program with the Apple II Computer. Here the rudiments of turning the computer on and setting it up so you can use the VisiCalc program effectively will be described. If you are already familiar with this computer, you might want to merely skim most of this chapter and proceed to the next.

Physical Components of an Apple II Computer

Every Apple II Computer has a system unit which includes a keyboard. The "computer" of the Apple II resides in the system unit. In order to use the VisiCalc program, the computer must have 48K (approximately 48,000 positions) of memory, and a display console and a diskette drive. The computer can have optional devices, such as a printer, as well as a second diskette drive. A diskette drive is also called a floppy disk drive, or simply a disk drive.

Figure 2.1 on the next page shows the display unit above the system unit with the keyboard, and it identifies important items. A disk drive can hold one five-inch diskette. Each diskette can record approximately 127,000 characters of information. Some of these characters may be used to represent computer programs such as the VisiCalc program and some of them are used to represent your data. The word "byte" is often used in place of the word "character"; for our purposes, these words are equivalent.

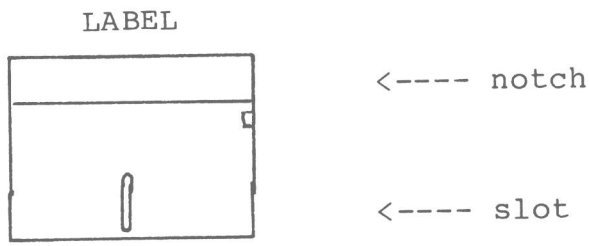
If you have two disk drives on an Apple II Computer, you can have immediate access to one-quarter of a million characters of information. If your computer has only one disk drive, it is called drive 1. If it has two disk drives, the one furthest from the computer is referred to as drive 2.

(©Allen Ruid, Photographer, 1982)

Figure 2.1 The Apple II Plus Computer

Using Diskettes

Diskettes are normally stored in protective jackets or sleeves and they should be stored in an upright position. When you are about to use a diskette you should remove its protective jacket. This will reveal a rigid black square protective envelope. Diskettes have an oval slot in the square envelope which protects the flexible (floppy) disk magnetic recording media, as shown below:



If you see a small notch on the edge, near the bottom right of the label, the information on that diskette can be changed and new information added. In such a case, we say that this diskette is "write-enabled." If on the other hand you see a piece of foil in the same position covering the notch, then the computer will refrain from changing any information on that diskette; the diskette is said to be "write-protected."

If you are inserting your one and only VisiCalc program diskette into the diskette drive, make sure its write-protect notch is covered. That will prevent accidental overwriting of your master copy. Your VisiCalc program diskette is identified as being "copy protected." That means you cannot duplicate it, so handle it with care.

When you insert the diskette into the diskette drive, the label side should be facing up and the edge furthest from the label must be inserted into the drive first. Gently push the diskette through the drive door slot. Push the drive door shut.

Each disk drive has an indicator light just below the disk drive slot, to the left, with a label "IN USE." The indicator light will glow whenever the disk drive is being used by the computer. You should not attempt to insert or extract a disk when this light is glowing. Doing so could destroy the diskette and damage the disk drive. If it seems that the indicator light won't ever stop glowing, it would be better to turn off the computer to perform this operation.

Some Preliminaries

The VisiCalc program is very easy to start up and use, as we shall see in a moment. Later on it will be helpful for us to make some use of the Apple II program known as the Disk Operating System DOS. The DOS reference manual has 200 pages. Don't be alarmed! We will discuss what we need to know about DOS in just a few pages, after we have gone further in using VisiCalc.

Operation of the Apple II Computer

We will walk through the steps required to start computing, and we will discuss the purpose of these steps, in case you are wondering about what is going on.

1. Place the diskette labeled "VisiCalc Program Diskette" in drive 1 (remember that drive 1 is the one which has its cable directly connected to the computer and for that reason it will usually be closer to the computer than drive 2).

2. Turn the computer on, by pushing the computer's on/off switch up. It is located on the left rear of the computer. Also turn on your display unit.

3. The computer will then begin reading information from the diskette. Whenever the computer is actively using a disk drive, the corresponding indicator light will glow; never remove a diskette while that light is glowing.

4. After a few dozen seconds, your display screen should greet you with a "blank worksheet" with the title "SOFTWARE ARTS" prominently visible at the top of your screen.

You have just completed "loading" the VisiCalc program. Loading is the process of transferring a copy of a program from a diskette into the computer's memory and once the copy of the program is in the computer's memory, that program starts directing the computer's operation. If you turn the computer off, the information in the computer's memory (also called RAM) will be lost. Fortunately you can retrieve a fresh copy of the program from the diskette.

Computer people have managed to invent more words or find new uses for old words than you would imagine possible, given that computers have only been around a few dozen years. One of the words you will run into is "boot" or "booting." These words are used as synonyms for loading a program.

When you wish to turn off the computer, it is wise to remove any diskettes that may still be in the disk drives. Remember that turning the computer off will erase whatever you have been typing

into the worksheet. So you may wish to save the worksheet on a diskette before you turn the computer off. How you do this and all the other things we have described in this chapter will occupy us for the next few chapters.

The Keyboard

Figure 2.2 shows you the keyboard we will be working with. It is very much like a typewriter keyboard, plus a few extra keys that will prove to be very useful as we go on.

(©Allen Ruid, Photographer, 1982)

Figure 2.2 Apple II Plus Computer Keyboard

You can correct an error while typing an entry by pushing the ESC key located in the second from the top row of the keyboard, at the far left. If you push the ESC key it will erase the last character you typed. You can then type the correct information. When the entry appears to be correct, push the RETURN key (the large key on the far right, second from the top).

In some situations you have to push the RETURN key to cause the VisiCalc program to examine your response, but as a general rule no special character will be displayed at the point you pushed the RETURN key. Note: whenever you see the word RETURN included among other items to be typed, simply push the RETURN key --do not type the 6 letters R, E, T, U, R, and N.)

Using Files

Almost anything you do with a computer involves using "files." A file is a body of information to which you have assigned a name and which you have recorded on a diskette. The VisiCalc program keeps track of which files are where by maintaining a directory of file names on each diskette. The