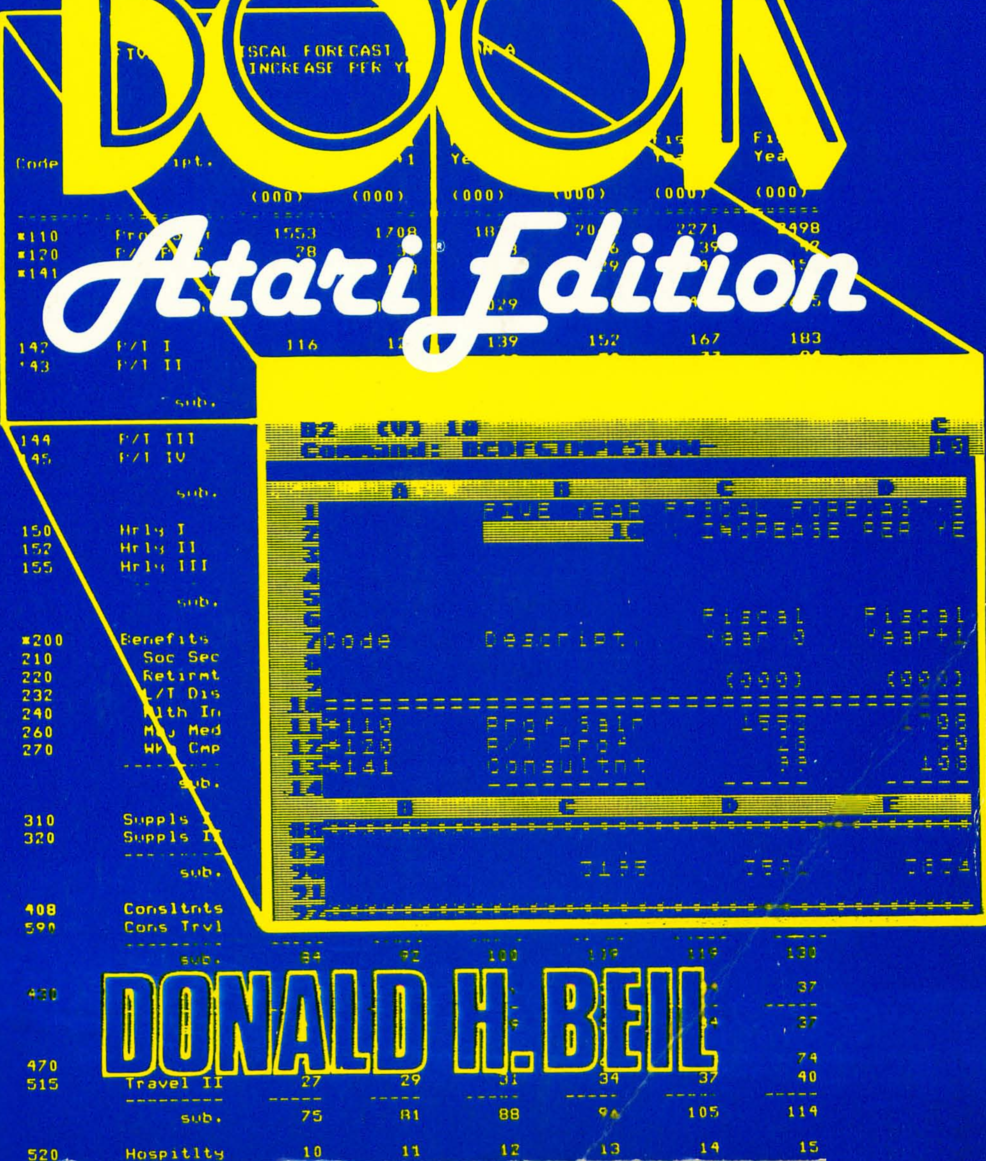
A REWARD BOOK

$14.95

THE VISICALC® BOOK

Atari Edition



DONALD H. BEIL

The VisiCalc® Book

Atari® Edition

The VisiCalc® Book

Atari® Edition

Donald H. Beil
National Technical Institute for the Deaf
Rochester Institute of Technology

Reston Publishing Company, Inc.
A Prentice-Hall Company
Reston, Virginia

Library of Congress Cataloging in Publication Data

Beil, Donald H.
The VisiCalc book, Atari edition

Bibliography
Includes index.
1. Atari computer--Programming. 2. VisiCalc (Computer program) I. Title
QA76.8.A82B44 001,64'25 82-5399
ISBN 0-8359-8394-3 AACR2
ISBN 0-8359-8393-5 (pbk.)

10 9 8 7 6 5 4 3 2 1

Printed in the United States of America

For my parents

Contents

Preface

This book has been an exciting project. In large part VisiCalc itself is responsible. Its creators are recognized in Chapter 1, Introduction.

VisiCalc, a computer program, is a tool that in a short time has achieved widespread use, extreme popularity, and significant praise for its usefulness and originality of design. In writing about its use, I hope to serve a variety of VisiCalc users:

New VisiCalc users will find a series of presentations that help the user become familiar with VisiCalc capabilities and manner of functioning. The book is organized to present a straightforward approach to using VisiCalc productively on a computer and to understanding the relationships between hardware, VisiCalc, spreadsheets, data, and the people who use this system. The large number of practice problems is included to help the new user become a skilled user. Many figures illustrate and explain how VisiCalc can be used.

Potential VisiCalc purchasers will find a presentation of both the capabilities and the limitations of using VisiCalc. This book can place this product in perspective and help in deciding if VisiCalc can be productive for solving problems for a potential user.

Data Processing managers who are responsible for evaluating and approving software purchases within their corporation or institution can use this book to evaluate the appropriateness of this product to meet the needs of those requesting it.

Experienced VisiCalc users will find presentations on creating templates, on preparing documentation, on user training, and on recognizing, preventing, and correcting errors. The book is also an indexed reference guide, which presents the features of VisiCalc. It provides a general orientation to producing VisiCalc sheets that are useful parts of the system that we use or may be used by our clients, secretary, or supervisor. A short training session is outlined; it can be used to train those who will work with the templates that we prepare.

Educators and trainers can use the book as a framework for coursework and training on the use of VisiCalc. The book provides an understanding of the power of VisiCalc and of the cautions necessary with this system. Practice problems are included; these can be assigned to students or trainees to help them in developing their skills.

Managers, clients, and others who use results obtained from VisiCalc models or who work with models prepared by others can use the book to develop an understanding of how those models are built and how reliable they are when used.

I want to recognize and thank Larry Benincasa of Reston Publishing Company for his immediate and continued enthusiasm and support of this project.

I discussed the book regularly with Dominic Fantauzzo of the National Technical Institute for the Deaf at the Rochester Institute of Technology (NTID/RIT) and benefited from his enthusiasm, suggestions, and interest. He, John Sweeney, and Paul Taylor each asked to read the manuscript and each offered valuable comments. Others at NTID/RIT who have helped or provided support include Rick Curwin, Robert Taylor, Lorna O'Brien, Warner Strong, William Castle, Nancy Fabrize, Barry Keesan, Mike Kleper, Sondra Milko, Bruce Peterson, Sheila Reasoner, Rosanne Rivers, Doug Sargent, and Alan Willett.

Philip F. Paul provided an important review of the manuscript, and I benefited from his suggestions.

Others who have been supportive or helpful, often in ways unknown to them, include Roz Beil; Sikandar Shaikh, C. R. Myers, and Nick Francesco, all of The Computer Center in Rochester, New York; William Carras; Frank Hacknauer; Marc Nodell; David Cole; and Al Tommervik.

My introduction to VisiCalc was through the product manual written by Dan Bricklin and Bob Frankston.

David Kroenke's framework for considering a computer system influenced my thinking and the organization of my writing in several chapters.

Many word-processing technicians have participated in preparing the manuscript. Although by choice they are usually not recognized by name, I want to do so. They are Laura Beiderbecke, Sharyn Bendzus, Dorothy Cerniglia, Petr J. Chudoba, Debra Dietch, Kathy Exner, Barbara Hall, Marcia Hood, Mary Jo Ingraham, Jane Johnson, Irene Kulesa, Tammy Marin, Theresa Northrup, Katrina Poquette, Betty Shaffer, Anita Sherman, and Gary Stape.

Ellen Cherry's work as production editor on this book for Reston Publishing Company was thorough and was done with great care. It was a pleasure to work with her again.

Others at Reston who have been helpful include Nikki Harden and Carol King.

My wife Marian and our sons Noah and Gabriel support me continually.

Donald H. Beil

Chapter 1

Introduction

Using VisiCalc® successfully involves an understanding of this product as part of a full system that includes not only VisiCalc itself but a number of other considerations as well. This environment or system includes

- The VisiCalc program we use.
- The computer on which we use the program.
- Uses to which we put VisiCalc, that is, the electronic sheets which we prepare and use.
- Data we enter when we use electronic sheets.
- People who use this system.

In this chapter, we'll briefly discuss the importance of each of these.

THE VISICALC SOFTWARE PROGRAM

VisiCalc is a computer program written by Dan Bricklin and Bob Frankston, Software Arts, Inc., which has it copyrighted, and is marketed by VisiCorp (Personal Software), Inc., and others. The program is sold as a package consisting of the VisiCalc program on a diskette, an accompanying manual, and a reference card, all packaged in a binder.

VisiCalc® is a registered trademark of Personal Software, Inc., VisiCorp.

Its capabilities are discussed generally in Chapter 2, The Capabilities of a VisiCalc System, and specifically throughout other chapters. Likewise, its limitations are discussed in Chapter 12, The Limitations of a VisiCalc System, and throughout the book. Other chapters describe how we use this system. A thorough understanding of what VisiCalc can and cannot do and how it is used is vital for us if we want to determine if and how it can be used to solve problems that we face.

All examples in the book have been prepared on an ATARI 800™ microcomputer. VisiCalc 1.74A has been used.

THE COMPUTER ON WHICH WE USE VISICALC

The computer, or hardware, on which we use, or run, VisiCalc will make a difference in how we can use it. For example, the size of the memory will directly affect our use of VisiCalc, since it can limit the problem solution we prepare.

THE USES FOR VISICALC

We will see that as we use VisiCalc to solve problems, we create what are called electronic sheets. To prepare them we'll need to know how to build these sheets. We'll discuss this in the following chapters: 3, Getting Started; 4, Commands; 5, Labels, Numbers and Formulas; 6, Built-in Functions; and 7, Other Topics.

In Chapter 9, Creating Templates, we'll discuss how to prepare models or patterns called templates. These are electronic sheets on which we've prepared models with some, but not all, of the values needed to calculate relationships. We'll complete these by entering the required values and rapidly obtaining our desired results.

We'll present ways of dealing with errors in our work in Chapter 8, Recognizing, Preventing, and Correcting Errors. Chapter 10, Documentation, provides some simple formats that we can use to record information of value to users (including ourself) of our work.

This area, using VisiCalc productively to solve our problems, is a major emphasis of this book.

THE DATA WE ENTER

With VisiCalc, as with any computer system, the results are heavily dependent upon the data provided, the numeric values and label information (for example, budget dollar amounts, employee names, etc.) that we enter. If we have an electronic sheet accurately prepared and then we enter data incorrectly, our results will most likely also be incorrect. We'll see that a VisiCalc system has limited capabilities for verifying the accuracy of data, a limitation that requires cautious use on our part. In Chapter 9, Creating Templates, we'll discuss this topic thoroughly.

SYSTEM USERS

If we want to use VisiCalc productively, we must be fully informed regarding its use. In addition, we'll find that others may use the VisiCalc electronic models, or templates, that we prepare. If others use our work, we must ensure that they are properly trained and have sufficient knowledge of their crucial responsibilities in the functioning of the full system. Chapter 11, What Our Client, Secretary, or Supervisor Needs to Know, contains an outline of a training session that we could conduct to ensure appropriate results.

Chapter 13, Practice Problems, is designed to provide a variety of problems to assist VisiCalc users in developing skills to use this system. The problems also suggest a wide range of potential uses of VisiCalc.

The Preface contains a short review of how other current or potential VisiCalc users may find this book useful.

Chapter 2

The Capabilities of a VisiCalc System

INTRODUCTION

VisiCalc is a powerful versatile software tool available for the ATARI 800 computer system. Its power lies in its advertised ability to provide the capabilities found in our use of a pencil, a sheet of paper, and a calculator. But because it provides an "electronic sheet," the power of the computer is combined with this software to give results accurately and readily and with great flexibility. It is useful in a wide variety of applications.

Budgeting and forecasting are two prime examples. In this chapter, we'll combine these two topics into one area, "budget forecasting," and discuss the capabilities of VisiCalc. In Chapter 12, The Limitations of a VisiCalc System, we'll examine the limitations of a VisiCalc-based system.

A SAMPLE PROBLEM: BUDGET FORECASTING

We'll begin our budget forecasting cycle with information about our current budget. Figure 2-1 shows our starting point. Notice that we've started with a simplified version of an expense budget; we'll build toward a more realistic example. The example of Figure 2-1 contains only a few lines with one column total. But even at this level, the complexities of budget forecasting can be demonstrated. PERSONNEL costs are dependent on the number of people employed (EMPLOYEES). BENEFITS are forecast as a percentage of PERSONNEL costs. TRAVEL expenses also depend on the number of people (EMPLOYEES).

As we sit down to do this forecast by hand, we have the budget worksheet of Figure 2-1 and some assumptions about the future. We expect to grow at 5 people per year, meaning that we'll go from 50 people in the current fiscal year (FY 0), to 55 next year (FY+1), to 60 in two years (FY+2), and finally to 65 people in three years (FY+3). Total PERSONNEL costs for the current year (FY 0) are calculated by multiplying the number of EMPLOYEES by the average salary of $20,000. Salaries are expected to rise 10% per year, with salaries for new employees calculated at the average salary for the year.

The computations of future PERSONNEL costs alone are not simple, as amounts are extended and increases for current employees are combined with costs for future new employees. However, once the PERSONNEL costs are determined, the BENEFITS computation is straightforward. In our example, BENEFITS are 25% of PERSONNEL costs. For the other lines, we'll need to make some assumptions about the future; for example, TELEPHONE, HOSPITALITY, and EQUIPMENT each increase at 10% per year, RENT is constant at $60,000 per year, and TRAVEL

```
                                BUDGET FORECAST
                  **************** YEAR *****************

                      FY 0        FY+1        FY+2        FY+3
                   *******     *******     *******     *******

 Employees              50


 Personnel         1000000
 Benefits           250000
 Telephone           10000
 Rent                60000
 Travel              40000
 Hospitalty           1000
 Equipment           18000
                  ---------

    TOTAL $        1379000
```

Figure 2-1. A budget forecasting worksheet for four fiscal years showing data for the current year (FY 0).

costs, currently budgeted at $800 per person, will increase 20% per year per person. We'll need to be careful with this line (TRAVEL) as it's also a function of the number of people working.

Figure 2-2 contains our budget, prepared by hand with a calculator for FY+1 through FY+3, following the rules established for ourselves in the preceding paragraphs.

Figure 2-2 is prepared by working carefully through the rows and columns and by recalling our assumptions as each item is calculated. Now suppose that we decide not to hire 5 new people per year but 6. Our hand calculations require a fresh worksheet, transfer of some numbers that we can carry from Figure 2-2 as is (although simply copying numbers introduces the potential for errors), and the recalculation of the PERSONNEL costs, BENEFITS, TRAVEL, and TOTAL lines; that is, everything affected by the number of EMPLOYEES.

Although we may get faster at the calculations, our tools are simply inadequate for the process. Our abridged budget of Figure 2-2 contains 27 values that we have entered or calculated (nine each for three years, including the totals). Most entries are a func-

BUDGET FORECAST

**************** YEAR *****************

	FY 0	FY+1	FY+2	FY+3
	*******	*******	*******	*******
Employees	50	55	60	65
Personnel	1000000	1210000	1452000	1730300
Benefits	250000	302500	363000	432575
Telephone	10000	11000	12100	13310
Rent	60000	60000	60000	60000
Travel	40000	52800	69120	89856
Hospitalty	1000	1100	1210	1331
Equipment	18000	19800	21780	23958
TOTAL $	1379000	1657200	1979210	2351330

Figure 2-2. Our budget worksheet with values calculated by hand for the next three fiscal years.

tion of some other entry as established by our own budgeting rules. If we want to change those values, we need to recalculate each entry dependent on its assumption.

Inadequate tools is an understatement. The tools available are so weak that we can be prevented from doing a thorough analysis. The effort can easily go into preparing a numerically accurate budget rather than a planned one.

THE POWER OF VISICALC

VisiCalc is an excellent tool for solving this problem and many others. Working at the computer, we can create a worksheet, often called an electronic sheet or spreadsheet, consisting of our budget line-item names, column headings, our FY 0 actual values, and then "formulas" for calculating all other entries based on our assumptions.

The simple budget used in earlier figures appears in part on the screen of a computer, as shown in Figure 2-3. Notice that we can see only part of the budget worksheet even though we have entered the full sheet. We'll discuss this limitation and ways VisiCalc provides to overcome it later in this chapter. Also notice that the worksheet on the screen follows the format in our earlier work.

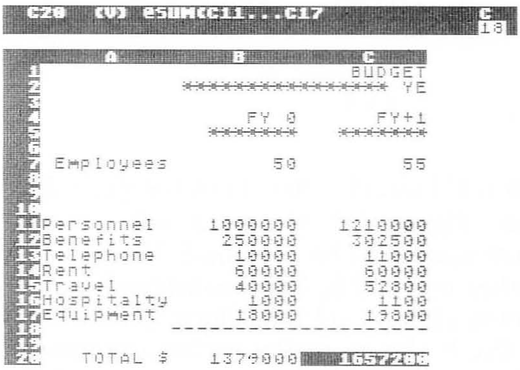


Figure 2-3. Part of our budget forecast on the screen of a computer using VisiCalc.

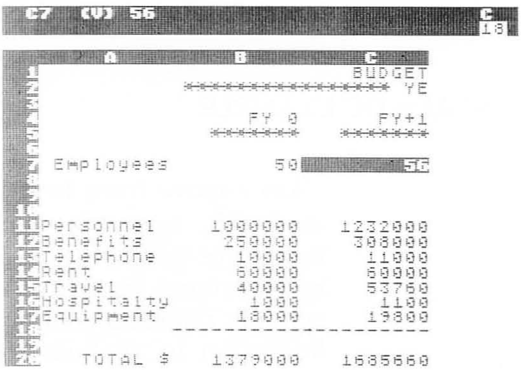


Figure 2-4. Our VisiCalc screen after revising the number of new EMPLOYEES, assuming that we hire six new people per year instead of five.

We have used VisiCalc to prepare the sheet in a format to match our desires; we are able to create these worksheets to meet our needs. (We'll discuss the information displayed on the border of the screen in Chapter 3, Getting Started.)

Now let's see how we can use this VisiCalc sheet as a replacement for the inadequate tools discussed earlier.

Suppose we want to hire 6 people instead of 5 in each of the three future years. We can ask this type of "what if..." question easily with VisiCalc. We can change the entries for EMPLOYEES to 56, 62, and 68 and in a second or so have the complete budget revised to include the new values as displayed in part on the screen, as shown in Figure 2-4. The values for the year FY+1 have been updated to reflect the new forecasts for the additional employees.

Similarly, we could ask: "What if" we change our BENEFITS percentage? "What if" we change the cost of TRAVEL per employee? Other similar "what if" questions could be asked, and with VisiCalc we immediately have the impact of such a change displayed for us.

This is the power of VisiCalc.

The preparation of this particular budget spreadsheet requires approximately the same amount of time as preparing the values of Figure 2-2 by hand (this will not be true for all problems). In this time, we can create all relationships that we've discussed. Doing this job with VisiCalc gives us an astounding advantage in our forecasting calculations. Once the relationships are established, we can change any one value or relationship (called a formula) and immediately see the impact of that new assumption.

A REALISTIC EXAMPLE

Let's move from here to a real budget rather than the preceding abridged, invented version. Figure 2-5 shows a full expense budget in its final form as we want it to be prepared. Values must be calculated for each FY other than FY 0. The unabridged budget actually contains 34 line items plus 14 subtotal lines, plus a total line—that is, 49 lines, not the 8 shown in the earlier example in this chapter. The "real" job requires a significant amount of time for the calculations when performed by hand. Also consider the impact if we wished to forecast for 10 years, not 5 as shown here.

Notice that the budget has line items, subtotals, and final totals; that it has underlines; and that there are assumed but invisible relationships between entries. For example, the BENEFITS may be

FIVE YEAR FISCAL FORECAST BASED ON A
10 % INCREASE PER YEAR

Code	Descript.	Fiscal Year 0	Fiscal Year+1	Fiscal Year+2	Fiscal Year+3	Fiscal Year+4	Fiscal Year+5
		(000)	(000)	(000)	(000)	(000)	(000)
*110	Prof.Salr	1553	1708	1878	2065	2271	2498
*120	P/T Prof	28	30	33	36	39	42
*141	Consultnt	99	108	118	129	141	155
	sub.	1680	1846	2029	2230	2451	2695
142	P/T I	116	127	139	152	167	183
143	P/T II	54	59	64	70	77	84
	sub.	170	186	203	222	244	267
144	P/T III	107	117	128	140	154	169
145	P/T IV	18	19	20	22	24	26
	sub.	125	136	148	162	178	195
150	Hrly I	62	68	74	81	89	97
152	Hrly II	15	16	17	18	19	20
155	Hrly III	0	0	0	0	0	0
	sub.	77	84	91	99	108	117
*200	Benefits						
210	Soc Sec	209	229	251	276	303	333
220	Retirmt	112	123	135	148	162	178
232	L/T Dis	12	13	14	15	16	17
240	Hlth In	51	56	61	67	73	80
260	Maj Med	15	16	17	18	19	20
270	Wkm Cmp	27	29	31	34	37	40
	sub.	426	466	509	558	610	668
310	Suppls I	44	48	52	57	62	68
320	Suppls II	62	68	74	81	89	97
	sub.	106	116	126	138	151	165
408	Consltnts	72	79	86	94	103	113
590	Cons Trvl	12	13	14	15	16	17
	sub.	84	92	100	109	119	130
430	Telephone	25	27	29	31	34	37
	sub.	25	27	29	31	34	37
470	Travel I	48	52	57	62	68	74
515	Travel II	27	29	31	34	37	40
	sub.	75	81	88	96	105	114
520	Hospitlty	10	11	12	13	14	15
544	Recruitng	36	39	42	46	50	55
580	Advsrs I	0	0	0	0	0	0
585	Advsrs II	0	0	0	0	0	0
	sub.	46	50	54	59	64	70
620	Equp Rent	57	62	68	74	81	89
	sub.	57	62	68	74	81	89
630	Repair	38	41	45	49	53	58
631	Maintnce	42	46	50	55	60	66
	sub.	80	87	95	104	113	124
*755	Contngncy	10	11	12	13	14	15
756	Contng756	10	11	12	13	14	15
	sub.	20	22	24	26	28	30
**912	Equipment	120	132	145	159	174	191
**914	Equip>200	104	114	125	137	150	165
	sub.	224	246	270	296	324	356
Total $		3195	3501	3834	4204	4610	5057

Figure 2-5. Our full budget with forecast information supplied for FY+1 through FY+5.

computed as a percentage of the salary line items. VisiCalc has all these capabilities. However, we are working at a computer system with a screen that has limitations on the number of lines and characters (numbers, letters, punctuation, blanks, etc.) that can be displayed simultaneously. Our entire real budget will not fit on the screen. This is a consideration when VisiCalc is used to solve problems.

However, VisiCalc has a series of commands to control which portion of the sheet appears on the screen. It is possible to "move" the sheet "under" the screen and therefore see parts of it as we desire. Some of these screen capabilities are discussed and demonstrated later in this chapter; all will be fully explained in the book.

SCREEN FORMATTING

VisiCalc offers an on-line window onto the electronic sheet. The sheet seems to move or "scroll" under the window. This action is like reading microfiche in a microfiche reader. The microfiche contains much more than we can see at one time. With VisiCalc, we can reposition the window so that it shows portions of the sheet, as in Figure 2-6. This shows the "screen" overlaid on the larger budget sheet. The window moves over the sheet like a magnifying glass.

VisiCalc, in its screen formatting capabilities, is much more powerful than a microfiche reader or a magnifying glass. For example, it can change the width of the columns shown. In this way, if we narrow the columns, we can "see" more of the sheet, as in Figure 2-7, where 6 columns appear on the screen instead of 4 in the previous figure.

These two figures show only numerical values; our column headings (top) and budget identification (left) have scrolled from view. Another capability of VisiCalc is a Title command that allows us to fix, or freeze, titles corresponding to what is on the remainder of the screen. Titles can be fixed vertically (columnwise), horizontally (rowwise), or in both directions. Figure 2-8 is similar to Figure 2-7 except that both the column labels and row labels are shown on the screen in correspondence to the numeric values shown.

In addition to these capabilities, the screen can be split horizontally or vertically into two windows each of which displays part of the sheet. Each portion can then be scrolled separately, and each can have separate titles and column widths. This is a powerful capability, as shown in Figure 2-9, in Figure 2-10, and again in Figure 2-11.

Figure 2-9 shows a horizontal split in the screen. We can observe the effect on the fiscal year (FY) totals at the bottom of each column while changing other parts of the screen, for example, the percentage increase per year in our budget. Figure 2-9 shows several of the fiscal year totals for a 10% increase per year in each line item starting in FY+1, and Figure 2-10 shows the same window for a 14% increase starting in that year. At our computer we can immediately see the impact on the total annual budget caused by this projected increase in our budget. The second window occurs within seconds after we change the percentage.

In the next illustration, we've selected a vertical split in the screen. In Figure 2-11, we see the effects on some of the line items of our FY+5 budget caused by the change to a 14% per year increase in our budget.

USING THE ELECTRONIC SHEET

Preparing the relationship between entries of the sheet involves creating formulas which actually become entries of the sheet. For example, look at the part of the sheet shown in Figure 2-12. We'll provide a brief description of the sheet here and a full description in Chapter 3, Getting Started.

First notice the columns and rows of information, like a large accounting sheet. Each column on the screen is labeled at the top. The columns now on the screen are B, C, D, and E. Each row also is labeled, but with numbers, not letters. We can see rows 26, 27, ... 44, 45 on the screen in Figure 2-12. Each place where a column and row meets is called an entry. Each entry is named with the column label, then row label, for example, A1, G35, or C29.

At any entry on the sheet, we can write a label, such as the word

BENEFITS

at entry B37. Or, we can write a numeric value such as the value

209

at location C38. Or we can write a relationship, called a formula, on the sheet at an entry, as we'll see. We write all this information by typing the characters on the keyboard of the computer.

FIVE YEAR FISCAL FORECAST BASED ON A
10 % INCREASE PER YEAR

Code	Descript.	Fiscal Year 0 (000)	Fiscal Year+1 (000)	Fiscal Year+2 (000)	Fiscal Year+3 (000)	Fiscal Year+4 (000)	Fiscal Year+5 (000)
*110	Prof.Salr	1553	1708	1878	2065	2271	2498
*120	P/T Prof	28	30	33	36	39	42
*141	Consultnt	99	108	118	129	141	155
	sub.	1680	1846	2029	2230	2451	2695
142	P/T I	116	127	139	152	167	183
143	P/T II	54	59	64	70	77	84
	sub.	170	186	203	222	244	267
144	P/T III	107	117	128	140	154	169
145	P/T IV	18	19	20	22	24	26
	sub.	125	136	148	162	178	195
150	Hrly I	62	68	74	81	89	97
152	Hrly II	15	16	17	18	19	20
155	Hrly III	0	0	0	0	0	0
	sub.	77	84	91	99	108	117
*200	Benefits						
210	Soc Sec	209	229	251	276	303	333
220	Retirmt	112	123	135	148	162	178
232	L/T Dis	12	13	14	15	16	17
240	Hlth In	51	56	61	67	73	80
260	Maj Med	15	16	[illegible]	[illegible]	[illegible]	[illegible]
270	Wkm Cmp	27	29	[illegible]	[illegible]	[illegible]	[illegible]
	sub.	426	466	[illegible]	[illegible]	[illegible]	[illegible]
310	Suppls I	44	48	[illegible]	[illegible]	[illegible]	[illegible]
320	Suppls II	62	68	[illegible]	[illegible]	[illegible]	[illegible]
	sub.	106	116	[illegible]	[illegible]	[illegible]	[illegible]
408	Consltnts	72	79	[illegible]	[illegible]	[illegible]	[illegible]
590	Cons Trvl	12	13	[illegible]	[illegible]	[illegible]	[illegible]
	sub.	84	92	[illegible]	[illegible]	[illegible]	[illegible]
430	Telephone	25	27	[illegible]	[illegible]	[illegible]	[illegible]
	sub.	25	27	[illegible]	[illegible]	[illegible]	[illegible]
470	Travel I	48	52	[illegible]	[illegible]	[illegible]	[illegible]
515	Travel II	27	29	[illegible]	[illegible]	[illegible]	[illegible]
	sub.	75	81	[illegible]	[illegible]	[illegible]	[illegible]
520	Hospitlty	10	11	12	13	14	15
544	Recruitng	36	39	42	46	50	55
580	Advsrs I	0	0	0	0	0	0
585	Advsrs II	0	0	0	0	0	0
	sub.	46	50	54	59	64	70
620	Equp Rent	57	62	68	74	81	89
	sub.	57	62	68	74	81	89
630	Repair	38	41	45	49	53	58
631	Maintnce	42	46	50	55	60	66
	sub.	80	87	95	104	113	124
*755	Contngncy	10	11	12	13	14	15
756	Contng756	10	11	12	13	14	15
	sub.	20	22	24	26	28	30
**912	Equipment	120	132	145	159	174	191
**914	Equip>200	104	114	125	137	150	165
	sub.	224	246	270	296	324	356
Total $		3195	3501	3834	4204	4610	5057

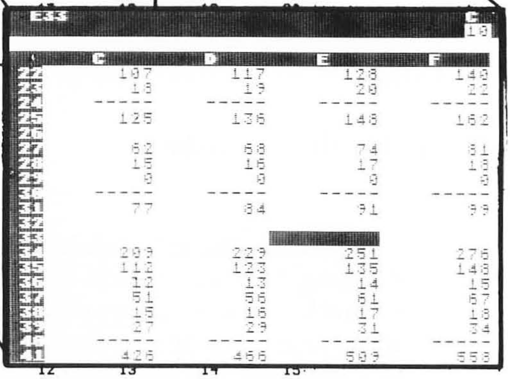


Figure 2-6. A screen overlaid on the budget sheet illustrating how VisiCalc provides a window onto the sheet.

FIVE YEAR FISCAL FORECAST BASED ON A
10 % INCREASE PER YEAR

Code	Descript.	Fiscal Year 0	Fiscal Year+1	Fiscal Year+2	Fiscal Year+3	Fiscal Year+4	Fiscal Year+5
		(000)	(000)	(000)	(000)	(000)	(000)
*110	Prof.Salr	1553	1708	1878	2065	2271	2498
*120	P/T Prof	28	30	33	36	39	42
*141	Consultnt	99	108	118	129	141	155
	sub.	1680	1846	2029	2230	2451	2695
142	P/T I	116	127	139	152	167	183
143	P/T II	54	59	64	70	77	84
	sub.	170	186	203	222	244	267
144	P/T III	107	117	128	140	154	169
145	P/T IV	18	19	20	22	24	26
	sub.	125	136	148	162	178	195
150	Hrly I	62	68	74	81	89	97
152	Hrly II	15	16	17	18	19	20
155	Hrly III	0	0	0	0	0	0
	sub.	77	84	91	99	108	117
*200	Benefits						
210	Soc Sec	209	229	251	276	303	333
220	Retirmt	112	123	[illegible]	[illegible]	[illegible]	[illegible]
232	L/T Dis	12	13	[illegible]	[illegible]	[illegible]	[illegible]
240	Hlth In	51	56	[illegible]	[illegible]	[illegible]	[illegible]
260	Maj Med	15	16	[illegible]	[illegible]	[illegible]	[illegible]
270	Wkm Cmp	27	29	[illegible]	[illegible]	[illegible]	[illegible]
	sub.	426	466	[illegible]	[illegible]	[illegible]	[illegible]
310	Suppls I	44	48	[illegible]	[illegible]	[illegible]	[illegible]
320	Suppls II	62	68	[illegible]	[illegible]	[illegible]	[illegible]
	sub.	106	116	[illegible]	[illegible]	[illegible]	[illegible]
408	Consltnts	72	79	[illegible]	[illegible]	[illegible]	[illegible]
590	Cons Trvl	12	13	[illegible]	[illegible]	[illegible]	[illegible]
	sub.	84	92	[illegible]	[illegible]	[illegible]	[illegible]
430	Telephone	25	27	[illegible]	[illegible]	[illegible]	[illegible]
	sub.	25	27	[illegible]	[illegible]	[illegible]	[illegible]
470	Travel I	48	52	[illegible]	[illegible]	[illegible]	[illegible]
515	Travel II	27	29	[illegible]	[illegible]	[illegible]	[illegible]
	sub.	75	81	88	96	105	114
520	Hospitlty	10	11	12	13	14	15
544	Recruitng	36	39	42	46	50	55
580	Advsrs I	0	0	0	0	0	0
585	Advsrs II	0	0	0	0	0	0
	sub.	46	50	54	59	64	70
620	Equp Rent	57	62	68	74	81	89
	sub.	57	62	68	74	81	89
630	Repair	38	41	45	49	53	58
631	Maintnce	42	46	50	55	60	66
	sub.	80	87	95	104	113	124
*755	Contngncy	10	11	12	13	14	15
756	Contng756	10	11	12	13	14	15
	sub.	20	22	24	26	28	30
**912	Equipment	120	132	145	159	174	191
**914	Equip>200	104	114	125	137	150	165
	sub.	224	246	270	296	324	356
Total $		3195	3501	3834	4204	4610	5057

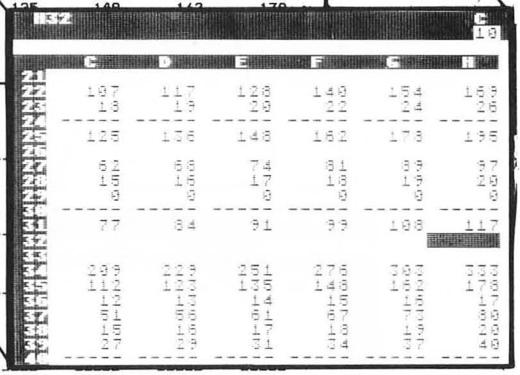
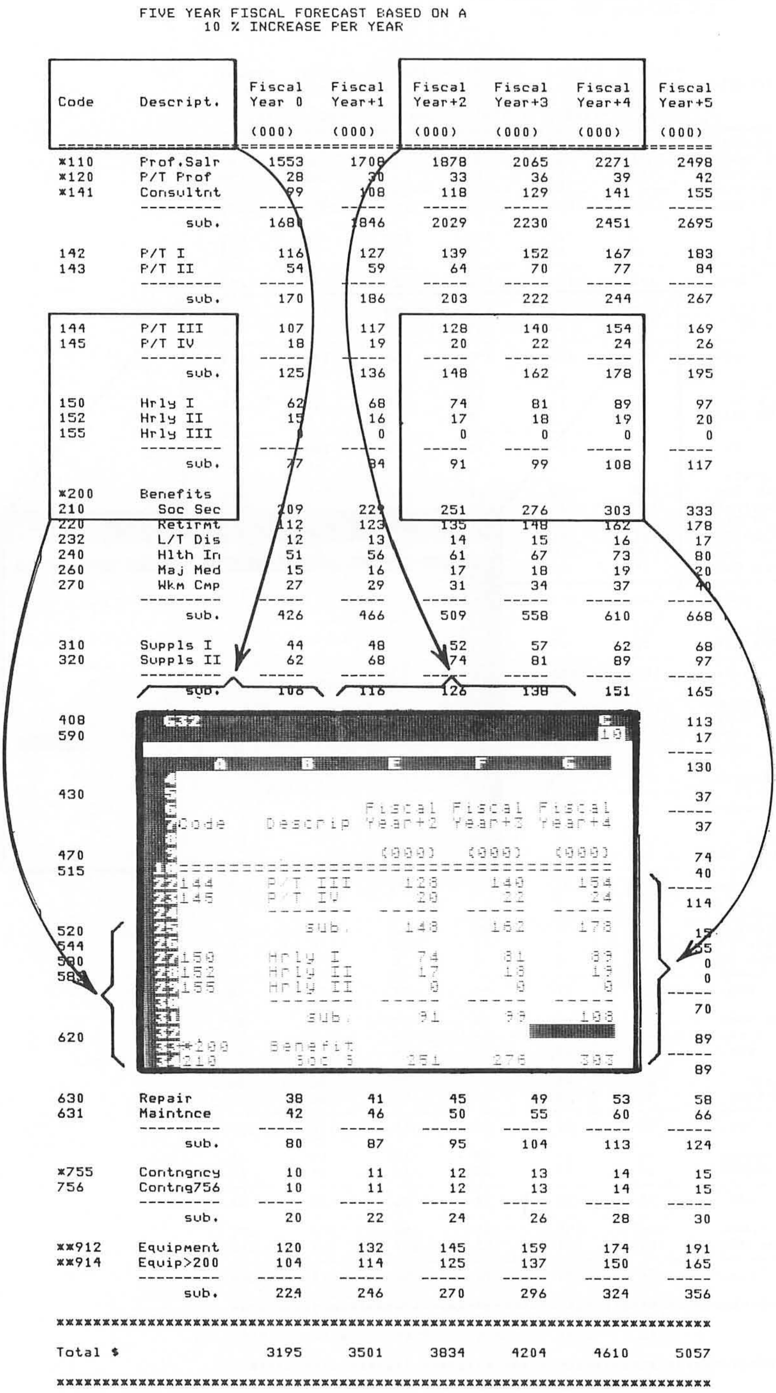


Figure 2-7. A window onto our budget with the budget displayed with narrower columns than in Figure 2-6.


FIVE YEAR FISCAL FORECAST BASED ON A
10 % INCREASE PER YEAR

Code	Descript.	Fiscal Year 0 (000)	Fiscal Year+1 (000)	Fiscal Year+2 (000)	Fiscal Year+3 (000)	Fiscal Year+4 (000)	Fiscal Year+5 (000)
*110	Prof.Salr	1553	1708	1878	2065	2271	2498
*120	P/T Prof	28	30	33	36	39	42
*141	Consultnt	99	108	118	129	141	155
	sub.	1680	1846	2029	2230	2451	2695
142	P/T I	116	127	139	152	167	183
143	P/T II	54	59	64	70	77	84
	sub.	170	186	203	222	244	267
144	P/T III	107	117	128	140	154	169
145	P/T IV	18	19	20	22	24	26
	sub.	125	136	148	162	178	195
150	Hrly I	62	68	74	81	89	97
152	Hrly II	15	16	17	18	19	20
155	Hrly III	0	0	0	0	0	0
	sub.	77	84	91	99	108	117
*200	Benefits						
210	Soc Sec	209	229	251	276	303	333
220	Retirmt	112	123	135	148	162	178
232	L/T Dis	12	13	14	15	16	17
240	Hlth In	51	56	61	67	73	80
260	Maj Med	15	16	17	18	19	20
270	Wkm Cmp	27	29	31	34	37	40
	sub.	426	466	509	558	610	668
310	Suppls I	44	48	52	57	62	68
320	Suppls II	62	68	74	81	89	97
	sub.	106	116	126	138	151	165
408							113
590							17
							130
430							37
							37
470							74
515							40
							114
520							15
544							55
580							0
58[illegible]							0
							70
620							89
							89
630	Repair	38	41	45	49	53	58
631	Maintnce	42	46	50	55	60	66
	sub.	80	87	95	104	113	124
*755	Contngncy	10	11	12	13	14	15
756	Contng756	10	11	12	13	14	15
	sub.	20	22	24	26	28	30
**912	Equipment	120	132	145	159	174	191
**914	Equip>200	104	114	125	137	150	165
	sub.	224	246	270	296	324	356
Total $		3195	3501	3834	4204	4610	5057

Figure 2-8. A window onto the budget sheet illustrating fixed titles.

FIVE YEAR FISCAL FORECAST BASED ON A
10 % INCREASE PER YEAR

Code	Descript.	Fiscal Year 0 (000)	Fiscal Year+1 (000)	Fiscal Year+2 (000)	Fiscal Year+3 (000)	Fiscal Year+4 (000)	Fiscal Year+5 (000)
*110	Prof.Salr	1553	1708	1878	2065	2271	2498
*120	P/T Prof	28	30	33	36	39	42
*141	Consultnt	99	108	118	129	141	155
	sub.	1680	1846	2029	2230	2451	2695
142	P/T I	116	127	139	152	167	183
143	P/T II	54	59	64	70	77	84
	sub.	170	186	203	222	244	267
144	P/T III	107	117	128	140	154	169
145	P/T IV	18	19	20	22	24	26
	sub.	125	136	148	162	178	195
150	Hrly I	62	68	74	81	89	97
152	Hrly II	15	16	17	18	19	20
155	Hrly III	0	0	0	0	0	0
	sub.	77	84	91	99	108	117
*200	Benefits						
210	Soc Sec	209	229	251	276	303	333
220	Retirmt	112	123	135	148	162	178
232	L/T Dis	12	13	14	15	16	17
240	Hlth In	51	56	61	67	73	80
260	Maj Med	15	16	17	18	19	20
270	Wkm Cmp	27	29	31	34	37	40
	sub.	426	466				
310	Suppls I	44	48				
320	Suppls II	62	68				
	sub.	106	116				
408	Consltnts	72	79				
590	Cons Trvl	12	13				
	sub.	84	92				
430	Telephone	25	27				
	sub.	25	27				
470	Travel I	48	52				
515	Travel II	27	29				
	sub.	75	81				
520	Hospitlty	10	11				
544	Recruitng	36	39				
580	Advsrs I	0	0	0	0	0	0
585	Advsrs II	0	0	0	0	0	0
	sub.	46	50	54	59	64	70
620	Equp Rent	57	62	68	74	81	89
	sub.	57	62	68	74	81	89
630	Repair	38	41	45	49	53	58
631	Maintnce	42	46	50	55	60	66
	sub.	80	87	95	104	113	124
*755	Contngncy	10	11	12	13	14	15
756	Contng756	10	11	12	13	14	15
	sub.	20	22	24	26	28	30
**912	Equipment	120	132	145	159	174	191
**914	Equip>200	104	114	125	137	150	165
	sub.	224	246	270	296	324	356
Total $		3195	3501	3834	4204	4610	5057

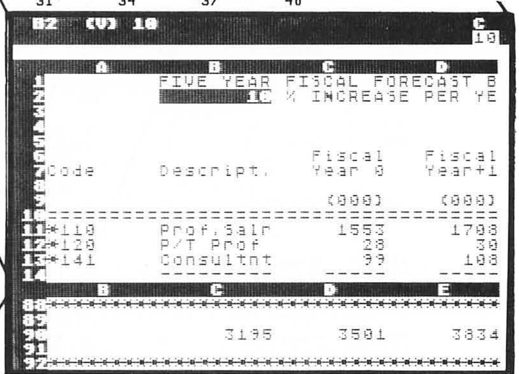
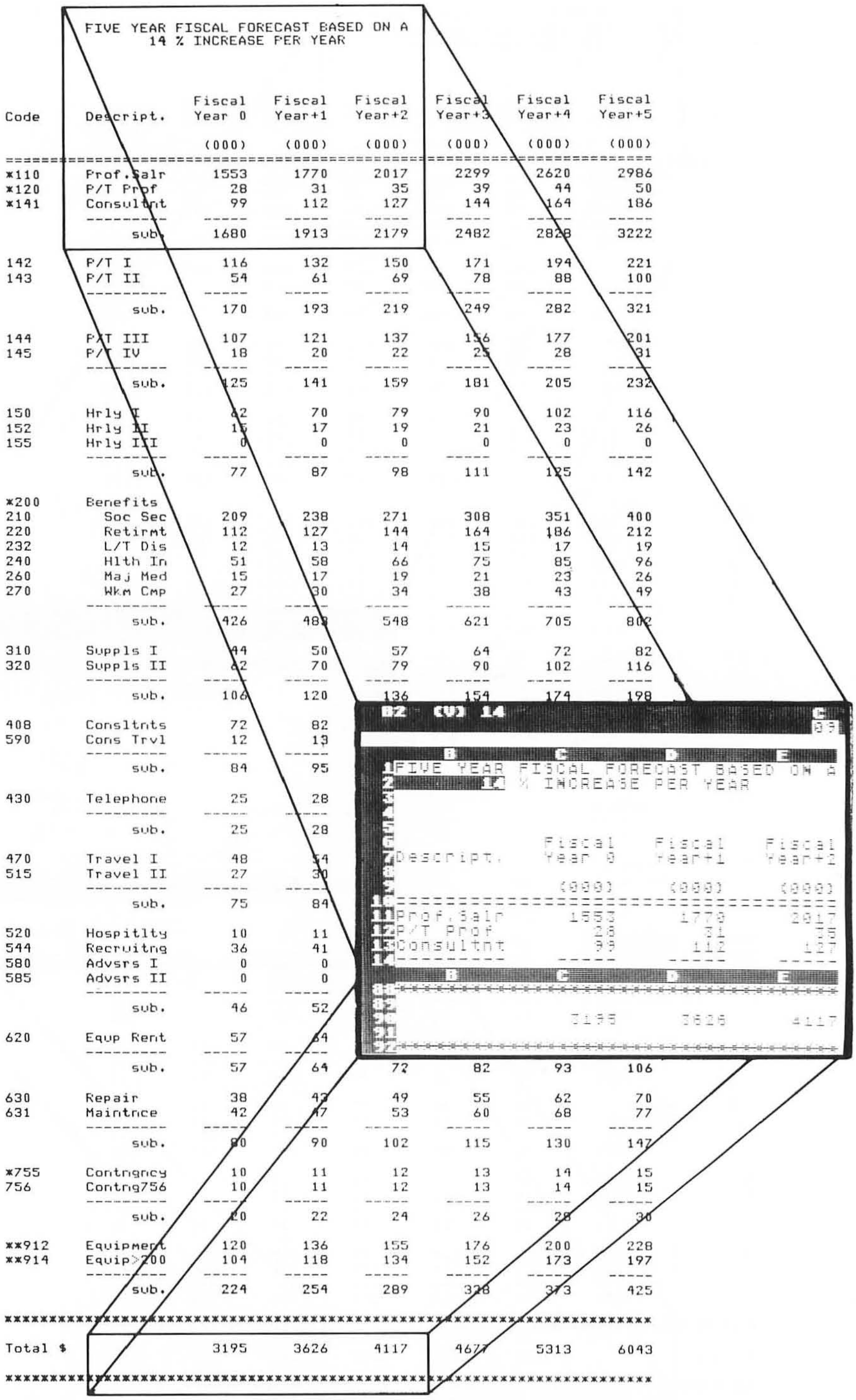


Figure 2-9. A horizontally split screen illustrating our totals (bottom) and another portion from the sheet (top).



FIVE YEAR FISCAL FORECAST BASED ON A
14 % INCREASE PER YEAR

Code	Descript.	Fiscal Year 0 (000)	Fiscal Year+1 (000)	Fiscal Year+2 (000)	Fiscal Year+3 (000)	Fiscal Year+4 (000)	Fiscal Year+5 (000)
*110	Prof.Salr	1553	1770	2017	2299	2620	2986
*120	P/T Prof	28	31	35	39	44	50
*141	Consultnt	99	112	127	144	164	186
	sub.	1680	1913	2179	2482	2828	3222
142	P/T I	116	132	150	171	194	221
143	P/T II	54	61	69	78	88	100
	sub.	170	193	219	249	282	321
144	P/T III	107	121	137	156	177	201
145	P/T IV	18	20	22	25	28	31
	sub.	125	141	159	181	205	232
150	Hrly I	62	70	79	90	102	116
152	Hrly II	15	17	19	21	23	26
155	Hrly III	0	0	0	0	0	0
	sub.	77	87	98	111	125	142
*200	Benefits						
210	Soc Sec	209	238	271	308	351	400
220	Retirmt	112	127	144	164	186	212
232	L/T Dis	12	13	14	15	17	19
240	Hlth In	51	58	66	75	85	96
260	Maj Med	15	17	19	21	23	26
270	Wkm Cmp	27	30	34	38	43	49
	sub.	426	483	548	621	705	802
310	Suppls I	44	50	57	64	72	82
320	Suppls II	62	70	79	90	102	116
	sub.	106	120	136	154	174	198
408	Consltnts	72	82				
590	Cons Trvl	12	13				
	sub.	84	95				
430	Telephone	25	28				
	sub.	25	28				
470	Travel I	48	54				
515	Travel II	27	30				
	sub.	75	84				
520	Hospitlty	10	11				
544	Recruitng	36	41				
580	Advsrs I	0	0				
585	Advsrs II	0	0				
	sub.	46	52				
620	Equp Rent	57	64				
	sub.	57	64	72	82	93	106
630	Repair	38	43	49	55	62	70
631	Maintnce	42	47	53	60	68	77
	sub.	80	90	102	115	130	147
*755	Contngncy	10	11	12	13	14	15
756	Contng756	10	11	12	13	14	15
	sub.	20	22	24	26	28	30
**912	Equipment	120	136	155	176	200	228
**914	Equip>200	104	118	134	152	173	197
	sub.	224	254	289	328	373	425
Total $		3195	3626	4117	4677	5313	6043

Figure 2-10. The screen of Figure 2-9 moments after revising our line-item percentage increase from 10% to 14% per year.

FIVE YEAR FISCAL FORECAST BASED ON A
14 % INCREASE PER YEAR

Code	Descript.	Fiscal Year 0 (000)	Fiscal Year+1 (000)	Fiscal Year+2 (000)	Fiscal Year+3 (000)	Fiscal Year+4 (000)	Fiscal Year+5 (000)
*110	Prof.Salr	1553	1770	2017	2299	2620	2986
*120	P/T Prof	28	31	35	39	44	50
*141	Consultnt	99	112	127	144	164	186
	sub.	1680	1913	2179	2482	2828	3222
142	P/T I	116	132	150	171	194	221
143	P/T II	54	61	69	78	88	100
	sub.	170	193	219	249	282	321
144	P/T III	107	121	137	156	177	201
145	P/T IV	18	20	22	25	28	31
	sub.	125	141	159	181	205	232
150	Hrly I	62	70	79	90	102	116
152	Hrly II	15	17	19	21	23	26
155	Hrly III	0	0	0	0	0	0
	sub.	77	87	98	111	125	142
*200	Benefits						
210	Soc Sec	209	238	271	308	351	400
220	Retirmt	112	127	144	164	186	212
232	L/T Dis	12	13	14	15	17	19
240	Hlth In	51	58	66	75	85	96
260	Maj Med	15	17	19	21	23	26
270	Wkm Cmp	27	30	39	38	43	49
	sub.	426	483	546	621	705	802
310	Suppls I	44	50	57	64	72	82
320	Suppls II	62	70	79	90	102	116
	sub.	106					
408	Consltnts	72					
590	Cons Trvl	12					
	sub.	84					
430	Telephone	25					
	sub.	25					
470	Travel I	48					
515	Travel II	27					
	sub.	75					
520	Hospitlty	10					
544	Recruitng	36					
580	Advsrs I	0					
585	Advsrs II	0					
	sub.	46					
620	Equp Rent	57					
	sub.	57	64	72	82	93	106
630	Repair	38	43	49	55	62	70
631	Maintnce	42	47	53	60	68	77
	sub.	80	90	102	115	130	147
*755	Contngncy	10	11	12	13	14	15
756	Contng756	10	11	12	13	14	15
	sub.	20	22	24	26	28	30
**912	Equipment	120	136	155	176	200	228
**914	Equip>200	104	118	134	152	173	197
	sub.	224	254	289	328	373	425
Total $		3195	3626	4117	4677	5313	6043

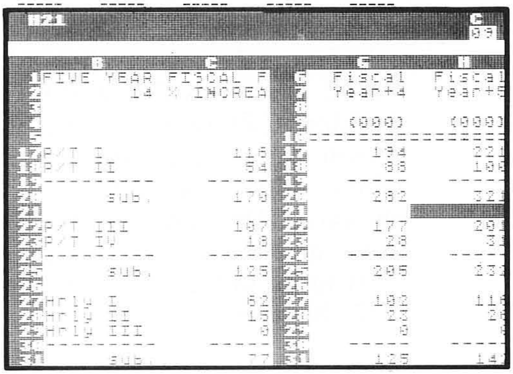


Figure 2-11. A vertical split in the screen illustrating the relationship between years FY 0, FY+4, and FY+5.

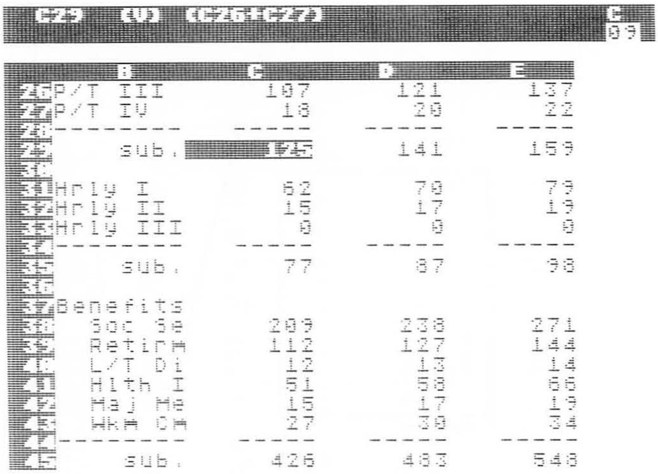


Figure 2-12. Part of the budget electronic sheet used to illustrate the establishment of formulas created to provide relationships between entries.

In Figure 2-12, the value at location C29 is the total of the values stored in entries C26 and C27. This formula itself is entered into position C29 of the sheet. The formula does not print on the screen as the entry; instead, its value (currently 125) prints. Then when values for C26 or C27 are entered or changed, C29 is changed. The formula at C29 remains the same, but its current value may change, depending on the contents of C26 and C27.

At location C29, we actually enter the formula relating C26 and C27. We type

(C26+C27)

at the keyboard. This provides our formula for this entry.

Let's look at another feature of VisiCalc. Look at column E. In our budget, the formulas that we want here are similar to all those in column D.

VisiCalc provides the capability, in a single step, of replicating these formulas down a column or across a row. We enter the formulas in one area of the sheet, for example, in column D. We then have the ability to reproduce these formulas automatically in columns E through H with one command. In doing so, the formulas can be replicated relative to their locations; that is, all summation formulas will add numbers from the correct columns (not from column D). This powerful command enables us to build and revise these relationships rapidly and easily.

BUILT-IN FUNCTIONS

VisiCalc includes approximately twenty "built-in functions," which can greatly simplify formula preparation. For example, at location C45 of the sheet in Figure 2-12, we can enter the formula

(C38+C39+C40+C41+C42+C43)

which is similar to the formula used at C29. However, we can also use the SUM built-in function (which is explained fully later in the book). This is a timesaver because with it we only have to enter

@SUM(C38.C43)

There are many built-in functions, including summation, averaging, counting, finding the maximum or minimum, and the net present value. There are also scientific functions such as logarithms (LN, LOG10, and EXP), sine, cosine, tangent, square root, arcsine, arccosine, arctangent, a value for pi, etc.

CREATING MODELS (OR TEMPLATES)

There are other features of VisiCalc, of which the capability to create templates is an extremely important one. Look at Figure 2-13. It's our budget again, but many of the entries show ..... instead of a value. For example, entry B5 contains a series of periods, and not a numeric percentage. Similarly, entries C14, C15, C16, and others show the periods and not numeric values. Many others show the value 0 (zero).

This is an example of a template. The dots are printed at locations at which values are to be entered. In creating this, we are providing a blank form for a user to complete. The user then fills in the blanks and, in doing so, obtains results for chosen values.

The steps needed to enter numbers in these positions can be taught to another person by the creator of the template. Thus, the function of entering the data can be performed by someone without the skill necessary to create the templates. This can be an extremely important timesaver and means that the template designer's skill can be used in preparing additional templates while a client, secretary, or supervisor uses the model.

The templates are reusable, meaning that once written, they can be used repeatedly or can be distributed to many units of an organization. For the experienced user who also performs the data entry function, the existence of the template is also important for

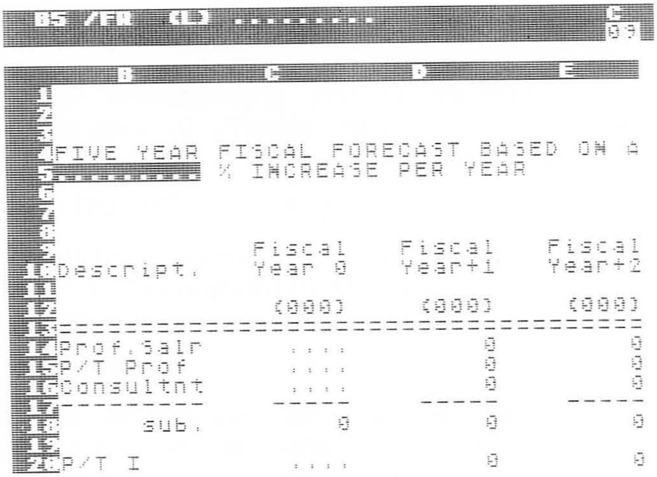


Figure 2-13. Our budget shown in model, or template, form.

many reasons, including use with different projections and the general flexibility that they provide.

ADDITIONAL CAPABILITIES

Other capabilities of VisiCalc include the ability to insert, delete, or move rows and columns.

Individual entries, or the entire sheet, can be formatted in a variety of ways, for example, values rounded to integers, or numbers shown with two decimal places (dollars and cents), etc.

All or part of the sheet may be printed.

We can prepare primitive bar graphs as shown in Figure 2-14, where the budget amounts from our forecast are shown graphically.

VisiCalc also has capabilities for storage and retrieval of sheets on diskettes or tape cassettes. This means that an application that has been placed on a spreadsheet can be reexamined as needed when conditions change or that reports can be generated periodically or on an as-needed basis. Since these data diskettes can be copied, multiple copies of the sheets and templates can be distributed to various VisiCalc users or can be stored as backup copies. At several places in the book, we'll discuss the importance of this backup capability.

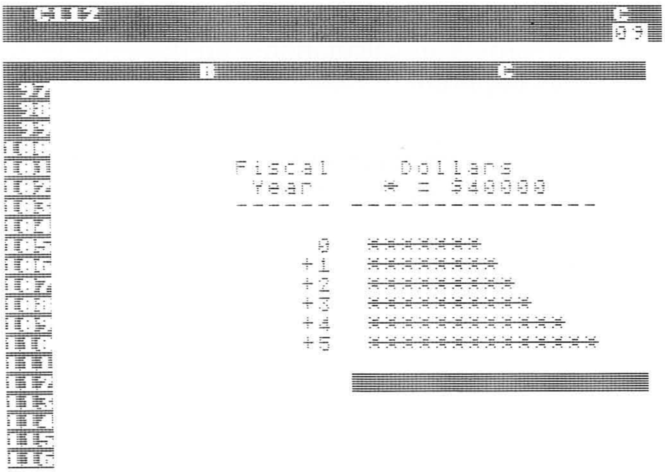


Figure 2-14. An illustration of the limited graphing capabilities of VisiCalc.

SUMMARY

This chapter has provided an introduction to the capabilities of VisiCalc:

- An electronic sheet on which we write labels (column or row titles, underlines, etc.), numbers, or formulas.
- The capability to recalculate values rapidly when new values are entered.
- A variety of screen formatting facilities to allow for the electronic sheet to scroll under the screen, for titles to be fixed in place over columns or next to rows, for the screen to be split horizontally or vertically, for column widths to be changed, etc.
- The ability to use built-in functions to simplify formula preparation.
- The capability to create templates that can be used like blank forms on which we write our data.
- Commands to insert, delete, move, and copy rows and columns.
- Formatting commands to control the display format of individual entries of the sheet or of all entries on the sheet.
- Commands to print all or parts of the sheet.

- The ability to store electronic sheets, in various stages or with a variety of calculations, on diskettes or cassettes for repeated use as required.

The remainder of this book provides the knowledge to use these capabilities, as well as to understand their limitations.

Chapter 3

Getting Started

INTRODUCTION

This chapter helps us to begin using VisiCalc to solve problems. The chapter assumes that the reader is able to load VisiCalc successfully into the memory of the system in use. This implies the ability to turn on the system, load a diskette properly into the disk drive, and take the steps necessary to load and run VisiCalc.

Those unable to complete these steps should refer to either the manuals accompanying the computer hardware or the VisiCalc manual. These steps become routine for a regular user, while an occasional user may at times require assistance from another user, a vendor, or a manual to help recall the necessary steps.

All readers who have a computer available are encouraged to follow the examples in this chapter, as well as throughout the book, on that system and to develop skills with VisiCalc by completing the exercises in Chapter 13, Practice Problems.

THE VISICALC WORKSHEET AND THE COMPUTER WINDOW

We're beginning with VisiCalc loaded properly into our computer and producing the screen like the one in Figure 3-1. Once this screen appears, the VisiCalc program diskette should be removed from the disk drive and replaced in its binder until the next time we want to use it. For now we'll ignore the information on the top three lines.

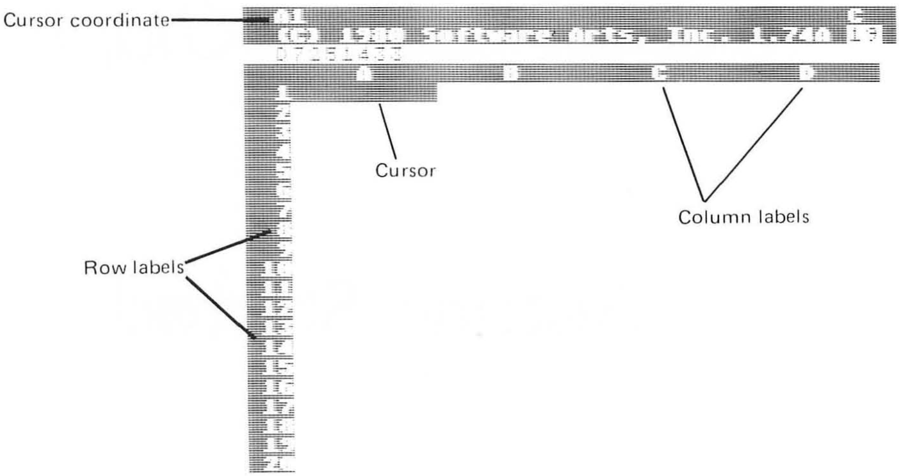


Figure 3-1. The screen after VisiCalc has been successfully loaded into the memory of the computer (refer to Chapter 1, Introduction, for details of the specific hardware and VisiCalc version used in this book).

VisiCalc provides us with a large worksheet, called an electronic worksheet, which has columns and rows named as follows: the columns are named with letters, and the rows with numbers. Notice columns A, B, C, and D, and rows 1, 2, 3, ... 19, 20 on the screen. These names, called labels, allow us to identify locations where columns and rows intersect.

Look closely at the intersection of column A and row 1, which is named, or labeled, A1 in VisiCalc. A1 is called the coordinate of that position on the sheet. Notice a rectangle at this position. This rectangle, pointing to A1, is called the cursor and acts like a highlight of one entry. On the keyboard, locate the arrow keys and the CTRL key; while watching the screen, we'll push the CTRL key, holding it down, and then push the key on which the arrow points to the right. When we do so, two things occur: first, some of the information on the top three lines of the screen disappears; and second, the rectangular cursor moves. The cursor is now positioned at the intersection of column B and row 1, called entry B1, or simply B1.

To use an arrow key on the ATARI, the CTRL key is pressed first, and while it is held down, the arrow key is pressed, then both are released. In this book we'll simply refer to pressing the arrow key and will recognize that this means use both the CTRL and arrow keys as described here. The ATARI keys have an auto-repeat capability, so that if the arrow keys (or other keys) are held down the key repeats its action.

Now depress the left arrow key, which returns the cursor to A1. Push the left arrow key again. This time the cursor flashes momentarily and an audible beep occurs. Nothing wrong has happened; VisiCalc has gently notified us that we have bumped into the left-hand edge of the sheet. Pressing the left arrow key repeatedly at this point will reproduce the same action.

Let's move right again, to B1, then C1, then D1, and then again, bringing us to E1. All of the column labels have changed and now instead of A, B, C, and D we have labels B, C, D, and E. Press the right arrow again several times (or hold it down) and notice that we are moving across the columns with new columns coming into view while others move off to the left. This action, called scrolling, can be reversed by pressing the left arrow key repeatedly, which returns us to the left.

As we move, notice that the top left corner of the screen, called the cursor coordinate, contains the name (or coordinate), of the entry at which the cursor points. This changes each time the cursor moves.

Move the cursor repeatedly to the right to location Z1, and then once to the right, revealing location AA1. Continued movement reveals AB1, AC1, ... AZ1, BA1, BB1, ... and finally we bump into the right-hand edge of the sheet at location BK1 (again with a flash of the cursor and a beep).

In total we have 63 columns on this worksheet, columns A through Z, AA through AZ, and BA through BK.

Let's return to A1 from our current location at BK1. We can do so by pressing the left arrow key repeatedly, or by using the auto-repeat capability (press a key and hold it down to repeat its function). The cursor moves quickly across the sheet, returning us to location A1.

Let's move down the sheet. To do so we'll use the arrow keys again.

Let's discover the bottom of the sheet by moving down from A1 with the "down" arrow key. The bottom occurs at row 254. Now from location A254 move horizontally across the sheet. (To do so, move the cursor with the right arrow key.) We find that entry BK254 is the bottom right corner of the sheet.

Suppose that we want to move into the middle of the sheet, for example, to G45. Do so using the methods that we've presented.

For practice we'll return to entry BK254 but do so without the arrow keys. VisiCalc provides a simple action allowing us to move the cursor directly from one entry to another. It's called the GO

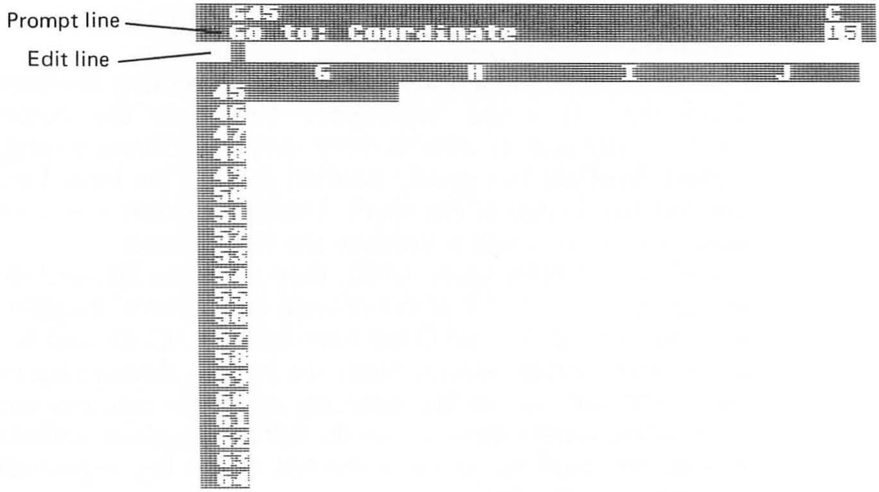


Figure 3-2. Our VisiCalc screen after we have begun the GO TO instruction but before entering the coordinate of the location we want.

TO. In order to use it, we first press the > key, after which the screen looks like Figure 3-2. Notice that the second line from the top, which had been blank, now says

GO TO: COORDINATE

This is the prompt line used by VisiCalc to communicate to us its expectations of our next action. Here we are being told that we have entered a GO TO and that VisiCalc is waiting for us to enter the COORDINATE of the location that we want.

Notice the box on the third line from the top of the screen. This is the edit line. In this situation, what we now type will appear on this line. We want to move to entry BK254. To do so, first type the letter B. The B appears on the edit line; the box has moved one position to the right, awaiting our next keystroke. Enter a K and notice that the line now has BK and the box. Enter the value 254 by typing the 2, then the 5, then the 4. At this point we have entered the desired coordinate. (If an error is made, read the next few paragraphs.) To indicate to VisiCalc that we're finished entering the coordinate, we press the key labeled RETURN. When we do so, the window changes completely and the cursor is now at location BK254 on the screen.

Let's return to G45 and deliberately type an error so that we learn one way of correcting typing errors with VisiCalc. Again we'll type the > key and enter our coordinate. Suppose that instead of entering G45, we type G456 and press RETURN. A beep occurs, not a move of the cursor. Our coordinate G456 is not on the sheet and nothing happens. In this instance VisiCalc did not prevent us from typing something incorrectly (which would be desirable) but did not accept the error.

Let's try again, entering the > followed by number 4 (without the G). This time VisiCalc does detect the error as we enter it; it issues a double beep and cancels the whole command.

Try again. This time enter

>G456

without the RETURN key. Suppose that we notice our error and simply want to erase the 6. To do so we use the key BACK S. Press this key and notice that it erases the last character entered. At this point we have our correct entry on the edit line and we press the RETURN key to complete the move.

The BACK S key can be used to erase more than one character. For example enter

>BK254 (no RETURN)

and then suppose that we want to erase all or part of this. Press the BACK S key repeatedly and notice how it removes what we have entered. Enter this same line again. Now press and hold down the BACK S key; notice that this erases, character by character, all that has been typed.

The same effect can be obtained by pressing the BREAK key. This fully cancels what we have entered, assuming that we have not yet entered the RETURN key.

Let's start again from location G45 and put the whole sheet into perspective by looking at Figure 3-3. Here we've superimposed the current screen over a representation of the entire worksheet.

The screen is really a window onto the electronic sheet. As we described in Chapter 2, The Capabilities of a VisiCalc System, this window is like a magnifying glass moving over the sheet at our direction or like a piece of microfiche in a reader. The sheet moves under the window, enabling us to view only a part of it at a time. However, as we'll see, our sheet is much more powerful than the microfiche, whose images cannot be changed.

Move the cursor back to entry A1 and let's start learning how to change the sheet by writing information of our own on it.

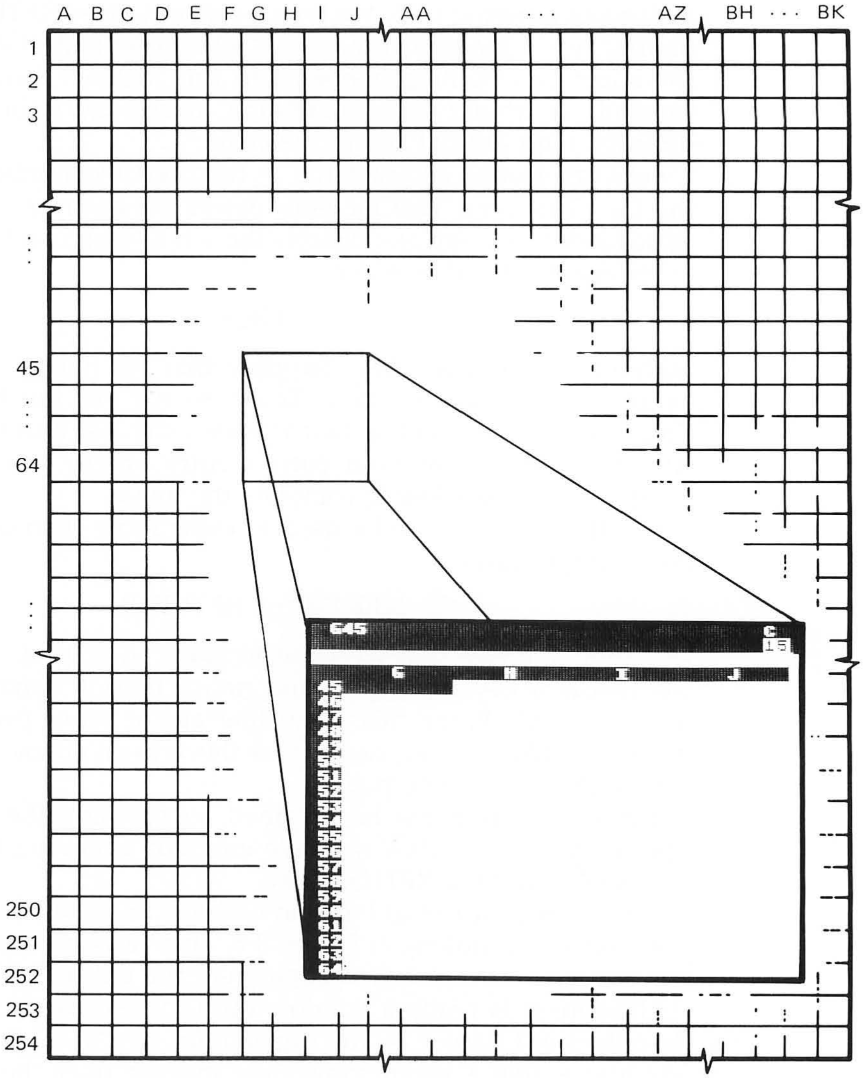


Figure 3-3. An example of a VisiCalc screen superimposed over a representation of the entire sheet.

WRITING ON THE ELECTRONIC SHEET

We will begin with four values that we want to total. Suppose that they are the number of units of a new product, which our company introduced in the FIRST quarter and then sold during the SECOND and THIRD quarter of the year. Now, the end of the third quarter, we are forecasting that we will sell 25 during the FOURTH quarter. We want to use VisiCalc to produce a report like the following.

QUARTER	UNITS SOLD
FIRST	8
SECOND	12
THIRD	15
FOURTH	25

TOTAL	60

In this section we'll introduce the steps necessary to prepare this report with VisiCalc. In the next section we'll expand this example to project our future sales and to demonstrate additional VisiCalc capabilities.

Let's start by noticing that this report really contains three different types of information:

- Column and row identification (QUARTER, FIRST, TOTAL, UNITS SOLD, ---------, etc.), called labels.
- Numbers (8, 12, etc.), called values.
- Finally, the TOTAL (60), a value but also the sum of the four quarterly values. This relationship, the sum, is called a formula.

We are able to begin writing anywhere on the sheet. Let's write our labels down the first column. Move the cursor to location A2, and we are ready to start entering our label, the word QUARTER.

When we type the first letter, Q, the prompt line shows the word LABEL, the Q followed by a box appears on the edit line, and the Q alone appears at the left of the cursor at entry A2. We will finish typing the word, and correct any errors we make with the BACK S key as we did earlier in this chapter. We'll use both upper and lower case letters, using the SHIFT for upper case as on a typewriter.

When we're satisfied with what we have entered, we press the RETURN key. A2 now shows the word QUARTER in the cursor; the top line of the screen, the entry contents line, shows

A2 (L) QUARTER

at the left of the line.

A2 is the coordinate of the cursor, (L) indicates that a label has been written at this position, and the word QUARTER is the label itself.

If we notice an error now, for example, a spelling error, we can enter the full label again and the old spelling will be removed and replaced by the new spelling.

Let's finish the column by moving the cursor down to A4. As we do so we see that we have left the word QUARTER at A2; that is, we have successfully written a label on the sheet. At A4 we'll enter the word FIRST. When it's correct, we'll push RETURN and then the down arrow. Now (at A5) type SECOND but don't press the RETURN key. Instead, we'll use the down arrow key. The action of the arrow keys includes the function performed by the RETURN key; therefore, both are not required. Push the down arrow key, and verify this result. Continue down the column, entering the words THIRD and FOURTH, then skipping row 8, and entering the word TOTAL at entry A9.

We have now finished column A. Use the arrow keys to move up and down column A, noticing how the entry contents line (the top row of the screen) repeats the contents of each entry where the cursor rests.

Move to B1 and enter the word UNITS, then to B2 and enter the word SOLD, and then to entry B4, where we are ready to enter our first value (as compared to the labels that we've entered so far). Here we press the 8 key, which places the word VALUE on the prompt line and an 8 followed by the box on the edit line (notice the value is not placed on the sheet yet). Now, as with a label, the RETURN or arrow keys are used to indicate that we're finished entering our value. Let's use the RETURN key and notice that the entry contents line shows

B4 (V) 8

and that the value 8 also appears at location B4. Moving down the column, we'll enter the other values, 12, 15, then 25.

For now we'll skip over the underline, and position the cursor at location B9. If we want, we can simply enter the value 60 at this point; however, we want the VisiCalc system to calculate this sum for us. If our UNIT SALES were not the simple values shown here or if our column listed monthly sales (12 lines) or daily sales (several hundred lines), the importance of allowing the computer to perform this calculation for us is clear. Remember also that our FOURTH quarter sales is a forecast, which we may want to change without the necessity of recomputing the total ourselves (more on this example later).

USING FORMULAS

At entry B9 we want VisiCalc to compute our sum. Here we'll enter our first formula. We want this entry to be the total of the values stored at entries B4, B5, B6, and B7. We'll examine two ways to do this.

One way to accomplish this is to write the full formula, the sum of the other four entries, at this entry. We'll write

(B4+B5+B6+B7)

as the formula. This means that at location B9 we will type the left parenthesis, at which time the word VALUE appears on the prompt line, then B, then 4, then +, then B, then 5, then +, and so on until we have entered the full formula, as shown in Figure 3-4. This figure shows the screen before pressing the RETURN key.

In typing this formula, a number of possible errors can be made, especially by those who are not regular computer users. In particular, in this formula, we may have problems in using the SHIFT key to get the parenthesis entered correctly. We may be beeped or discover that we have not entered what we wanted. If this occurs, use the BACK S key to correct the formula or use the BREAK key; if the formula has been accepted incorrectly (if it appears on the top line), then simply enter it again at the same location. Doing this will destroy the old incorrect formula and replace it with the new one.

If we enter the formula as we want and if we then type the RETURN key, two things occur:

- The formula appears on the entry contents line at the top of the screen.
- More importantly, our sum, 60, appears at location B9.

The screen, after the RETURN key is pressed, appears as shown in Figure 3-5.

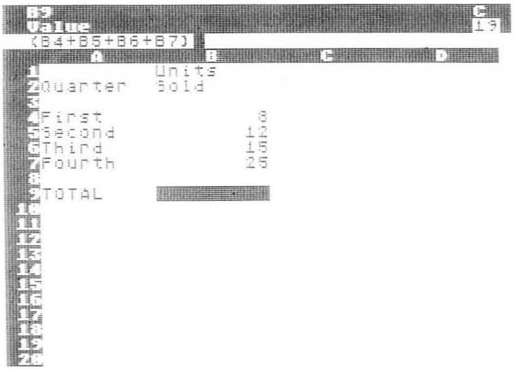


Figure 3-4. We have entered the formula for our total at location B9 but have not yet pressed the RETURN key.

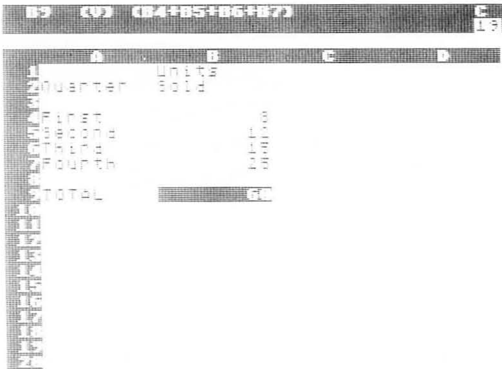


Figure 3-5. After pressing the RETURN key of the sheet in Figure 3-4, VisiCalc accepts and lists our formula on the top line (entry contents line) and displays the value of this formula, 60, at entry B9 of the sheet.

We have successfully entered our formula and have learned how to enter labels, numbers, and formulas.

It is significant to recognize that when a formula is stored, the current computed value of the formula is displayed. On the sheet we see the value 60, and not the long formula. This is different from the other entries on this particular sheet, where the item stored was identical to the item displayed. When we study formatting, we'll learn that we can control how items are displayed.

To illustrate the significance of this, let's suppose that we want to change our forecast for our FOURTH quarter entry: suppose that we want to forecast sales of 35, not 25. We'll move the cursor up to B7 and enter

35 RETURN

The results are shown in Figure 3-6, where our displayed total has changed from 60 to 70.

If we wish, we can continue this forecasting process, determining in turn the results of a series of "what if..." projections for the FOURTH quarter.

USING ONE OF THE BUILT-IN FUNCTIONS

Let's return the cursor to location B9 and examine a second way of writing our formula for this sum. Recall the possibility that our

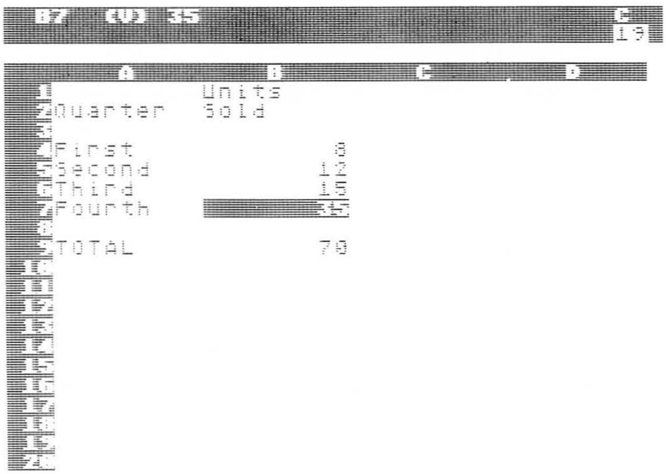


Figure 3-6. Our electronic sheet after changing the FOURTH quarter UNITS SOLD to 35 (instead of 25 as shown in Figure 3-5). Notice that the total is instantly recomputed and displayed as 70 (not 60).

record-keeping period may be monthly or daily instead of quarterly as shown here. If that is true, it means that our formula for the TOTAL will become extremely long, and therefore both time-consuming and difficult to enter accurately.

VisiCalc provides another way to total a column of numbers, called the built-in SUM function. We'll use it here, simply writing it on top of the old entry and thus replacing the previous formula.

We begin by entering the five characters

@SUM(

the first of which, the @, indicates that we want to use one of the VisiCalc built-in functions, and the next three, SUM, indicate the particular function that we want, here the SUM function. In Chapter 6, Built-in Functions, we'll learn that there are many others available in VisiCalc, and we'll study how each one is used.

At this point, after the left parenthesis we want to request a sum for the "series" of entries from B4 through B7. To do so, we'll first type

B4

then a single period, at which time VisiCalc expands the single period into three periods (...), an ellipsis, on the edit line (all for a visual effect). We then type

B7)

of which the B7 ends the series and the right parenthesis ends the built-in function. A RETURN gives us the screen of Figure 3-7.

Let's illustrate how this entry can be provided in a somewhat more convenient manner. Now we'll type only

@S(

to begin the sum built-in function. The edit line shows

@SUM(

even though we have not typed the UM. VisiCalc has printed these two middle letters for us on the screen.

At this point we're ready to enter the coordinate of the first entry of our series. To do so, we typed the location; however, here we'll demonstrate another way of doing this by pointing the cursor to the desired location. Press the up arrow. As we do so, the box on the edit line moves to the right; B8, the current cursor location, appears on the edit line. Push the up arrow again and observe that B8 changes to B7. Push it again and we have B6; again, B5; and finally, B4. If we pass B4, simply press the down arrow key to return to location B4. With B4 correctly on the line, press the period key (.). Three things occur: first B4 becomes a part of the edit line; second, B4 is followed by the ellipsis (...); and third,

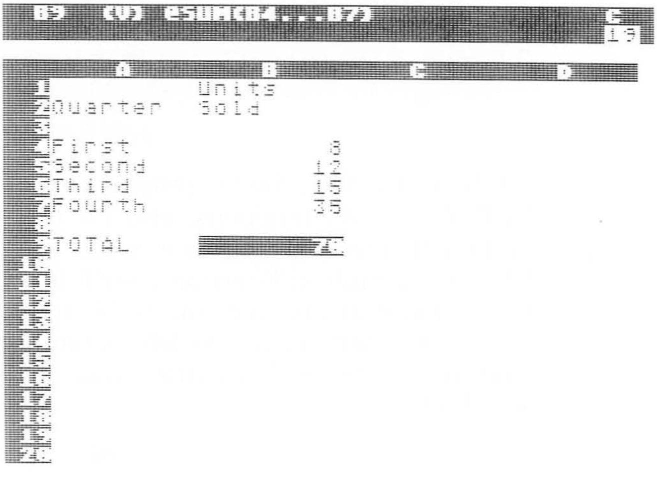


Figure 3-7. Entry B9 illustrates the use of the built-in SUM function to calculate the sum of the entries from B4 through B7.

the cursor has jumped back to position B9, where we are building the formula.

Let's complete the series by moving the cursor to B7, using the same technique used for B4. Again we'll press the up arrow key several times until the cursor is positioned at entry B7. Now we'll type

)

which, like the preceding period, results in B7 being accepted on the edit line, followed by the right parenthesis, and also causes the cursor to jump back to B9, our original location. The RETURN key completes the action.

The use of the cursor to point to entries that we want included in our formulas is a powerful tool with many applications.

With the SUM function, as with this formula, we can change any or all of the values in locations B4, B5, B6, and B7, and our numeric value displayed at B9 will be instantly updated.

In developing this sheet, we have used only a miminum of the available capabilities of VisiCalc; in fact we have not used any of the commands available. We will introduce a number of these in the next section.

SOME OF THE VISICALC COMMANDS

We'll continue with our quarterly report and present a number of the commands available with VisiCalc. Move the cursor to location B1. Notice that the label UNITS is at the left of this column, and not to the right of this column over the numbers. Let's introduce the commands of VisiCalc by using one of them to move this word to the right of this entry. First press the

/

key, and notice, as shown in Figure 3-8, that the prompt line shows

COMMAND: BCDFGIMPRSTVW-

We are being prompted to enter one of the many letters (or the hyphen) following the word COMMAND on the line. Each separate character represents one of the VisiCalc commands. *Some* of these are described below.

Character	Means
B	Blank the single entry at which the cursor is located.
C	Clear the entire sheet so that we can begin a new example with a clean sheet.
F	Provide a particular display format for the entry at the cursor location.
G	Take an action that affects the entire sheet, a global action.
I	Insert a row or a column.
R	Replicate one part of the sheet in another location on the sheet.
S	Use diskette (or cassette) storage in some way.
T	Freeze titles in place on the screen.
W	Split the screen to display part of the sheet in one area of the screen and another part in a separate area.
-	Use a repeating label.

We'll briefly demonstrate these. All are fully explained in Chapter 4, Commands.

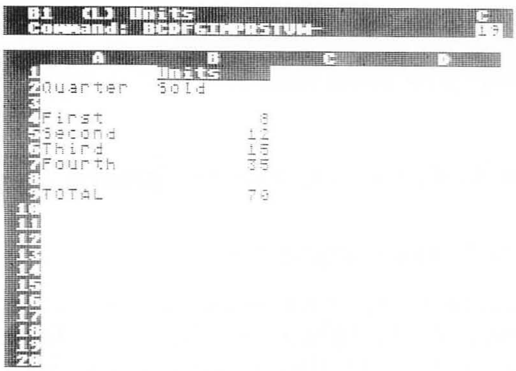


Figure 3-8. The screen after pressing the / key to begin a VisiCalc command.

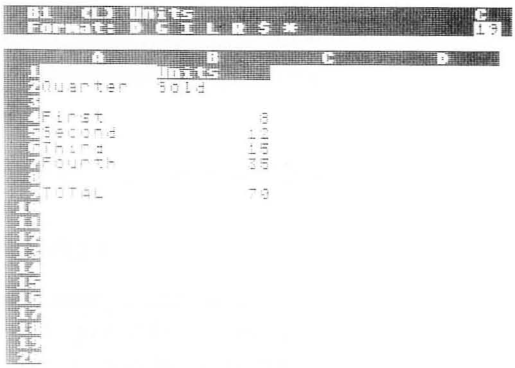


Figure 3-9. The same screen after typing F for the Format command.

THE FORMAT COMMAND

Let's start with the Format command. Press the F key, and notice the new prompt line

FORMAT: D G I L R $ *

which appears as in Figure 3-9.

Again we are presented with a series of choices (they are not called commands at this point), some of which have the following meanings.

Character	Means
I	Display this value as an integer. Notice that an I as a *command*, after a slash, has a different meaning. In one usage I means the Insert command, and in another usage I means integer. This is true of many characters in VisiCalc.
L	Display the item to the left of the entry, called left justified.
R	Display the entry at the right of the column, called right justified.

Right justification is what we want at this point, as we want to move the word UNITS to the right of the column. We press the R and observe the screen shown in Figure 3-10.

The entry contents line (the top line of the screen) now includes the characters

/FR

indicating right justification for this entry; the entry itself, the word UNITS, has been moved to the right. Move the cursor to position B2, and repeat the process, now by simply typing

/FR

in one sweep, while mentally thinking

Character	Means
/	Command
F	Format
R	Right justification

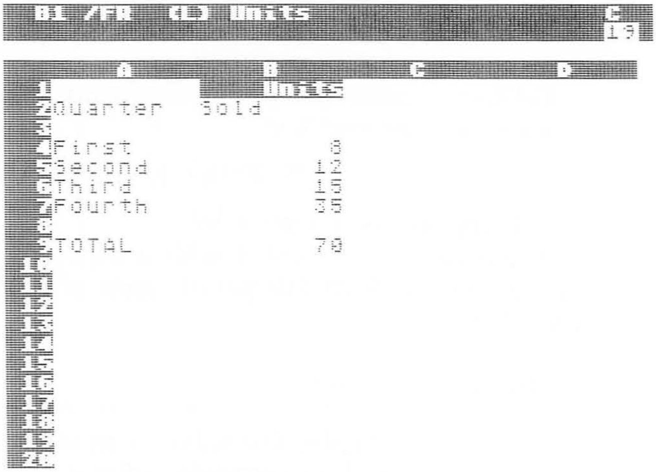


Figure 3-10. After typing an R, the word UNITS has been right justified in the field.

THE REPEATING LABEL COMMAND

Let's try another command. Move to location B8, which is blank. We'll use the Repeating Label command to obtain an underline at this entry. This is the - command, which is initiated by typing

/-

at which time the prompt line displays the line

LABEL: REPEATING

and the box appears on the edit line. Enter one underline character and a RETURN, that is,

_ RETURN

which fills entry B8 with an underline.

THE BLANK COMMAND

To remove the underline, we can use another command, the Blank command. Type

/B RETURN

and the underline disappears.

Let's compare the Format and Blank commands and notice an important distinction between the two. We typed /FR for the Format command and /B RETURN for the Blank command.

The Blank command required a RETURN while the Format did not. In general, VisiCalc requires a RETURN or other confirming action before completing an irreversible action. After the Blank command operates, we cannot retrieve what has been removed. However, the Format command is reversible, and as we'll learn later we can easily delete the format without affecting the contents.

Now at location B8 type

/-= RETURN

and notice that a row of double lines composed of equal signs appears.

USING LABELS

Let's suppose that we like the appearance of this double line but want to have only four equal signs right justified in the entry instead of the full row of nine now shown. We want to obtain a final result like the screen of Figure 3-11.

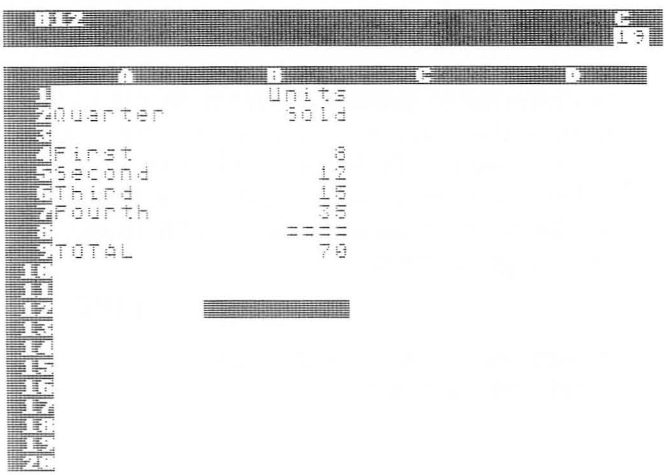


Figure 3-11. We'll add the double underline as shown at location B8.

By typing an indication that we want to right justify the field and that we want four equal signs, that is,

/FR==== RETURN

we think we should obtain the results we want. Let's try it. The first three characters are accepted as satisfactory; however, as we press the equal sign, it is not accepted and we are instead repeatedly beeped. In this instance VisiCalc is not sure of our action. It believes that we are entering an improper character and beeps us. To override this we must begin our entry with a double quote indicating that a label is being entered. Here entering

"==== RETURN

creates the desired result. Notice that the double quote is used at the start of our keystrokes, and not at the end.

We'll find that the double quote is required for starting labels that do not begin with a letter of the alphabet. This is discussed in Chapter 5, Labels, Numbers, and Formulas.

Another example is necessary to clarify the action of the double quote. Suppose that at our labels for the quarters we wanted to type

1ST	instead of	FIRST
2ND	instead of	SECOND
3RD	instead of	THIRD
and 4TH	instead of	FOURTH

When we type the 1 of 1ST, VisiCalc believes that we are entering a numeric value, not a label. As we type the next character, S, we are beeped, and the S is not accepted. In many instances this is desirable to prevent us from an error. For example, if we want to type the value 10 but instead of the digit zero we type the letter O (oh), we will be beeped and an error will be avoided.

However, here we are not in the process of making an error. As we learned, if we enter the double quote, our characters they will be accepted. Here we'll enter

"1ST RETURN

which causes all three of our characters, 1ST, to be accepted as the label that we want.

THE INSERT COMMAND

As we look again at the report we have been building, suppose we realize that it would be desirable to have the word PROJECTIONS appear on the first row of the report, above the current column labels. Let's demonstrate how to add it. Move the cursor to position B1 (any entry on row 1 will be satisfactory). Type

/I

to initiate the Insert command. This places

INSERT: R C

on the prompt line. Press the R to insert a row (the C inserts a column). As we do so, the entire report is shifted down one row. Repeat this action; that is, type

/IR

and it shifts down again.

Before we add our report heading (the word PROJECTIONS), move the cursor to location B11 and observe

@SUM(B6...B9)

on the entry contents line. The series, which had previously been

B4...B7

has been updated to accommodate the new locations of our quarterly values for UNITS SOLD. This is an extremely powerful feature, which allows us significant flexibility in building our sheets.

Let's return to row 1 and add the title that we want, the word PROJECTIONS. We want to center the word over the report, which means that we'll need to split it between column A and column B. At location A1 we'll type

/FR PROJE RETURN

and at location B1 we'll type

CTIONS RETURN

the rest of the label. This type of action, although often awkward, is commonly performed with VisiCalc. VisiCalc does not have another method of performing this label splitting.

THE GLOBAL COMMAND

Our sheet has the appearance shown in Figure 3-12. Let's change that appearance again, this time by displaying the sheet with columns that are narrower than the 9-character wide columns displayed here. Why do we change the width? There may be several reasons; for example in our problem we might want to move the two columns closer together to make them easier to read. We'll discuss another reason shortly.

We'll enter

/G

to initiate the Global command and indicate that we want to take an action affecting the entire sheet. The location of the cursor is unimportant with this command. The prompt line displays

GLOBAL: C O R F

We'll press the C, which places the words

COLUMN WIDTH

on the prompt line. Now we must decide what width we want and enter that width at this point. Let's enter a width of 7 by typing

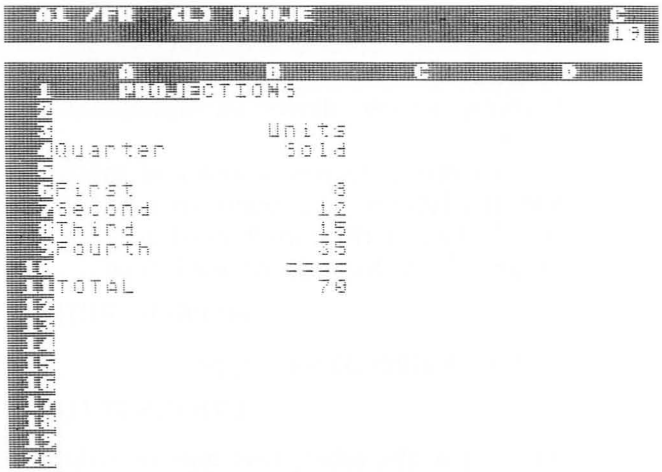


Figure 3-12. We've added two rows at the top of our sheet and centered the word PROJECTIONS between columns A and B by breaking the word between the two columns.

7 RETURN

All our report still appears, although it has been compressed to the left; significantly, column E (here all blank) appears on the screen.

We see another reason for using narrow columns; where appropriate we can increase the part of the sheet displayed on the screen.

Let's narrow the columns again, to a width of 5, by typing

/GC5 RETURN

Figure 3-13 shows the resulting screen. We observe a number of things. We've again increased the number of columns (here to 7); that is, columns A though G appear. We've lost parts of our labels. For example, at A4, where the cursor is located in the figure, the letters QUART and not QUARTER appear. However, look at the entry contents line that shows

A4 (L) QUARTER

This is important. We're displaying something different from what is stored on the sheet. We'll discover that many of the VisiCalc commands deal with the way information is displayed and not with the way it is stored. When we change the display format, we do not change what is stored on the sheet. Let's demonstrate this by returning the column width to 9, by typing

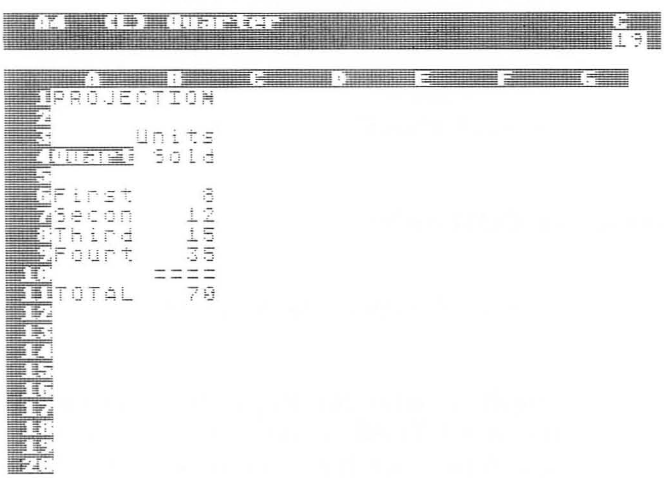


Figure 3-13. Our sheet after narrowing the column width to five characters.

/GC9 RETURN

which returns our complete report.

It is also important to observe that establishing a column width for a spreadsheet before entering labels will avoid the problems with label display encountered here.

ADDING FORECASTING TO OUR SALES REPORT

The report that we've worked on shows our activity for this year but not for future years. Let's add some forecasts, for example, for each quarter for the next 9 years, making this a 10-year report, including this year. We'll start by revising our column titles slightly. Let's blank locations B3 and B4 (position the cursor at each one separately and type /B RETURN). As we do so, notice that the format we entered earlier, /FR, remains. This is satisfactory for our purposes. (We'll learn how to erase it in the discussion of the Format command in Chapter 4, Commands). At location B3 enter the word YEAR and at location B4, the value 1. We've produced a simple column label

```
YEAR
   1
```

over the values in column B. Using the Replicate command, let's take the steps necessary to enter

```
YEAR  YEAR  YEAR       YEAR
   2     3     4 ....    10
```

over columns C, D, E, ... K.

THE REPLICATE COMMAND

Move the cursor to B3, and type

/R

which initiates the Replicate command. We'll use it to reproduce the word YEAR, instead of typing it nine additional times. It will also reproduce the associated format (here /FR), which again saves us from repeating this entry nine times. When we look at the screen now, we see a prompt line and an edit line as

REPLICATE: SOURCE RANGE OR RETURN
B3

Press the RETURN. This changes the two lines to

REPLICATE: TARGET RANGE
B3...B3:

At this point we have entered a SOURCE RANGE, which is the location of the item that we want to replicate. The SOURCE RANGE is the area from which we want to copy some items. In our example the location is B3...B3, which is a redundant way of stating that we want to reproduce the single item in location B3. (The redundancy is explained later). What's the target; that is, where do we want to reproduce this value? We want it from C3 through K3. To indicate this we'll type the now familiar "series" entry

C3.K3

in which the single period becomes an ellipsis (...). We have now created an edit line that contains

B3...B3: C3...K3

followed by a box.

Our SOURCE RANGE is to the left of the colon, and our TARGET RANGE is to the right. The TARGET RANGE is the area onto which we want to reproduce the items of the SOURCE RANGE. Watch the screen, and press the RETURN key. The word YEAR appears in location C3 and D3, visible on the screen, and in locations E3, F3, G3, H3, I3, J3, and K3, which we can verify by moving the cursor to these locations.

Let's move down to the next row, row 4, and discuss how we will replicate our year values, to produce 2, 3, 4, ... 10 in the desired locations. Place the cursor at location C4, and enter the formula

(B4+1) RETURN

which places the value 2 at this position on the sheet. This formula says that the value placed at C4 should be one greater than the value at location B4. That is, C4 should be equal to B4+1 as shown on the preceding line.

This seems awkward, when simply entering the number 2 would have done the same thing. We'll see why this formula is useful when we replicate again; however, instead of a label, we'll replicate a formula. With the cursor at location C4, type

/R

which gives us prompt and edit lines

REPLICATE: SOURCE RANGE OR RETURN
C4

Again we'll enter RETURN, now changing the two lines to

REPLICATE: TARGET RANGE
C4...C4:

Our target starts at D4, which we can type or which we can enter by cursor move. Let's do the latter, moving the cursor to D4. This is the same type of cursor move used earlier in the chapter. It gives us lines

REPLICATE: TARGET RANGE
C4...C4:D4

Pressing the period adds an ellipsis after the D4 on the edit line, as the cursor jumps back to C4. Let's locate the end of our target by cursor move, pressing the right arrow key until we are at position K4, under the last occurrence of the word YEAR. Now press the RETURN key, and study the screen in Figure 3-14.

The prompt line indicates with the word REPLICATE that we are not done with the Replicate command. We are being prompted to enter either an N, meaning No Change, or an R, meaning Relative. Notice that the entry line has been compressed and reads

C4: D4...K4: (B4

and that the coordinate B4 appears in a box. Finally notice that the entry contents show that the cursor location, C4, has the formula (B4+1) stored in it. Press the R key. The R here means relative. Most of the top three lines of Figure 3-14 disappear, and the values 3, 4, 5, ... 10 appear where we want them to appear in the target range of D4 through K4.

What happened? To explain it, move the cursor to D4, then E4, then F4, etc., and watch the formula stored at each location. It changes from (B4+1) at location C4, to (C4+1) at location D4, to (D4+1) at location E4, etc. The coordinate in each formula has changed *relative* to the entry.

If instead of an R we typed an N, we would have replicated the formula (B4+1) in each entry with No Change, meaning that our years would have been incorrectly labeled

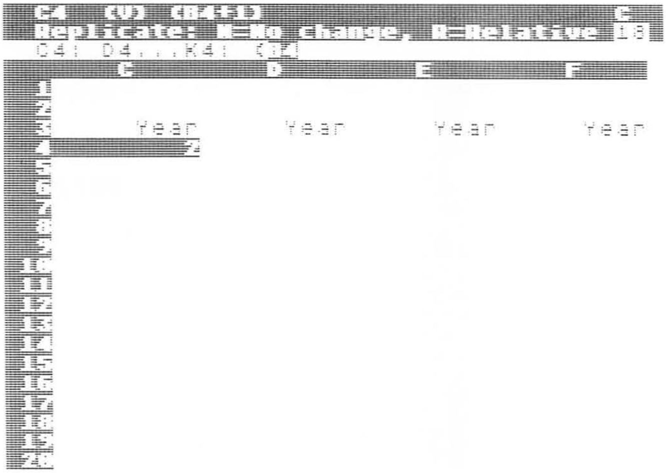


Figure 3-14. The screen as we are almost finished replicating the year value relatively across row 4.

1, 2, 2, 2, ... 2

and not

1, 2, 3, 4, ... 10

as we wanted. We'll explain this command in a number of additional examples in the book in Chapter 4, Commands.

In our example we are ready to include the quarterly forecasts for each of the next nine years. We need to supply the assumptions for the forecast: VisiCalc cannot do that. Let's make a simple assumption, that UNIT SALES will increase by 10% over the same quarter of the previous year. We have a business in which sales are seasonal, and we want to know what to expect for the quarters of the coming years.

At location C6 of Figure 3-15, we have entered the formula

(B6*1.1)

using the * to indicate a multiplication and using 1.1 to indicate a 10% growth; that is, C6 is equal to 110% of B6, or B6 times 1.1. Location C6 shows a value of 8.8, and our formula appears on the top line of the screen.

Let's replicate this formula down the C column into a target range of C7 through C9. This is identical in concept to the process that we followed to label the years 1, 2, 3, ... 10.

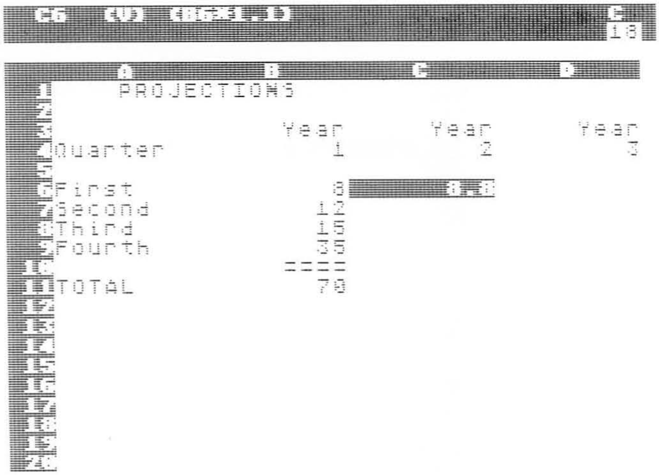


Figure 3-15. The beginning step of forecasting sales for each quarter of the next nine years.

We'll place the cursor at location C6 and then enter the following keystrokes:

/R:C7.C9:R

Let's explain each item that we've typed.

Keystrokes	Purpose
/	Enter the command structure of VisiCalc.
R	Select the Replicate command.
:	End the SOURCE RANGE, meaning that we're replicating from the sole entry C6. We can also use the RETURN key as we did in the earlier example in this chapter. The two function similarly here.
C7	Enter the first coordinate of the TARGET RANGE. A cursor move can also be used to specify this entry.
.	Indicate a series. The ellipsis (...) appears on the edit line.
C9	Enter the end of the TARGET RANGE. Again the cursor can be used.

:	Indicate the close of the series. As mentioned, the RETURN key also can be used.
R	Indicate that the coordinate in the formula is to be relative; that is, each quarter is 10% greater than the quarter of the previous year, not of the first quarter of year 1.

Now let's turn on the true power of the Replicate command by replicating the relationships that we've just established in column C across all the remaining eight years of our report. With the cursor again at C6 we type

/R.C9:D6.K6:RRRR

Let's explain again.

Keystrokes	Purpose
/	Enter the command structure of VisiCalc.
R	Select the Replicate command.
.	Indicate that the ellipsis is required for the middle of a series of our SOURCE RANGE. This is different from our earlier examples. Instead of a single entry as the source, we now have a series, beginning with C6. We'll see the end with the next keystrokes.
C9	Enter the coordinate of the end of the series.
:	Close this series (the RETURN key can also be used). We now have as our SOURCE RANGE a portion of a column rather than a single entry.
D6	Enter the start of the TARGET RANGE.
.	Indicate the middle of the target.
K6	Enter the end coordinate of the TARGET RANGE. Notice that our TARGET RANGE is across row 6 while our SOURCE RANGE is down column C. When this occurs—that is when the source and target areas are perpendicular—VisiCalc replicates the source onto a two dimensional rectangular area rather than onto a row or column as illustrated earlier. This is a powerful tool.
:	End the series of the TARGET RANGE.

R	Indicate Relative for the formula being replicated across row 6 (as explained in the next paragraph).
R	As directly above except for row 7.
R	As directly above except for row 8.
R	As directly above except for row 9.

The last four R's entered indicate that the formulas being replicated across this rectangle are to have a relative coordinate in their formulas. For these formulas it means that the percentage increase for each year will be 10% over the previous year, and not over the first year.

Let's move the cursor to the right to see some of the values that we've placed on the sheet. Figure 3-16 shows YEAR 7 through YEAR 10.

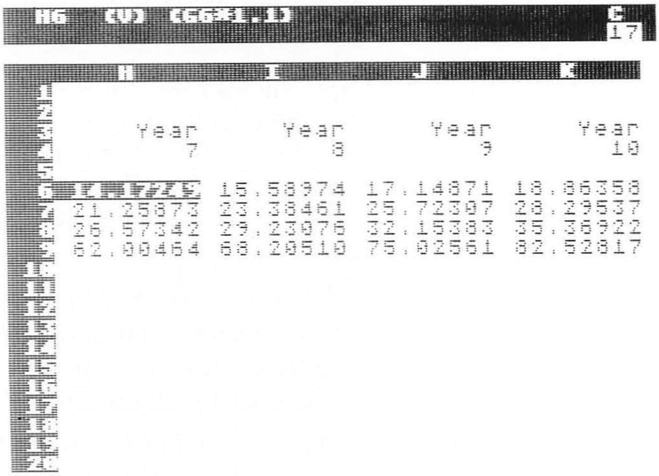


Figure 3-16. Our forecast of quarterly sales for YEAR 7 through YEAR 10.

We notice that the results are shown with five decimal places for all the entries here. Let's use the Global command to reduce the number of decimal places, first to two places and then to none. We'll enter

/GF$

and notice that these few keystrokes change all values to two decimal places as explained next.

Keystrokes	Purpose
/	Enter the command structure.
G	Select the Global command, whose action affects the entire sheet.
F	Select the format option.
$	Select the dollars and cents option from the formats available. Here we do not have dollar values but are using this option to obtain two decimal places.

After entering these keystrokes, we have the screen shown in Figure 3-17.

Notice that unfortunately we have also changed the column labels of row 4 to two places. We now have YEAR 7.00, YEAR 8.00, etc. To correct this, we can place the cursor at each year number, enter the Format command, and indicate that an integer, I, format is desired.

Instead of doing this only for the individual year labels, let's change all values to integer. To do so, we enter

/GFI

where all keystrokes have the preceding explanations except for I, which indicates integer format. When we enter these keystokes, our screen is again promptly revised and appears as in Figure 3-18.

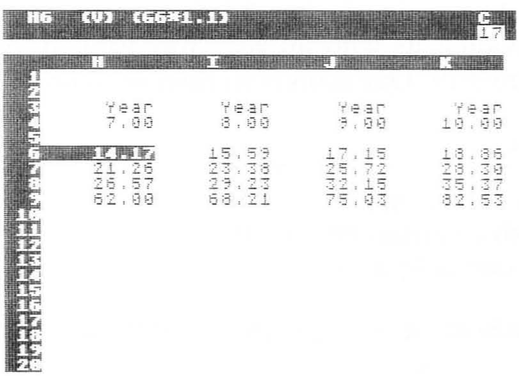


Figure 3-17. The forecast with values printed with two decimal places (a dollars and cents format).

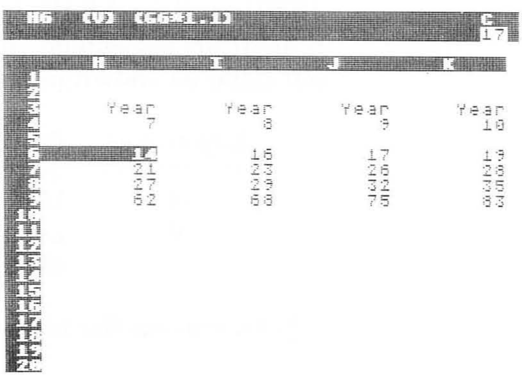


Figure 3-18. The forecast shown with an integer format for each entry.

We'll return the cursor to location B10, and replicate the double underline and the SUM function of column B across the other columns by entering

/R.B11:C10.K10:RR

In this case we did not need the intermediate step of entering formulas in column C. These two lines contain entries that can be replicated as is. The keystrokes RR at the end indicate that the series of the SUM function is Relative; that is, each SUM function at the bottom of a column sums the entries of that column.

Look carefully at the total for YEAR 2 (see Figure 3-20). It is not correct, as the sum of the four numbers shown is 78, not 77 as shown. VisiCalc, at our direction, is displaying these values as integers; however, the values stored on the sheet have several decimal places. In the SUM function, and in all formulas, the values used are the values stored on the sheet, not necessarily the values displayed on the screen. In this example, if we want the totals to be correct, we'll need to use the integer function, which is @INT and which we'll describe in Chapter 6, Built-in Functions.

THE TITLES COMMAND

Scroll to YEAR 10 and notice that the labels at the left move off the screen, which can make it difficult to identify our values correctly. Return to A6 and enter

/TV

Then scroll to YEAR 10. As we do so, our labels in column A remain on the screen, providing us with an easy way to identify our data, as shown in Figure 3-19. Our actions worked as follows:

Keystrokes	Purpose
/T	Use the Title command.
V	Establish a vertical title at, and to the left of, the cursor location.

Let's remove the titles, indicating no (N) titles by entering

/TN

which returns us to where we were.

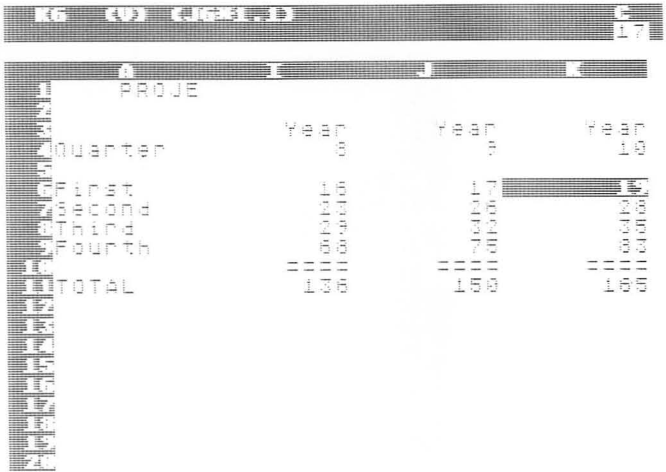


Figure 3-19. A demonstration of the use of the Title command to freeze titles at the left of the screen.

THE WINDOW COMMAND

If we want to see all our forecast on the screen, we can try to narrow the columns, for example, by scrolling to A6 and then entering

/GC3 RETURN

to obtain a global column width of 3. The results of this are shown in Figure 3-20. Notice that the totals for YEAR 5 through YEAR 10 are no longer on the screen; instead we see

>>

at each total location. In this example the sum is too large to fit into a column three characters wide (and leave a blank at the left to separate it from the next column). When this occurs VisiCalc prints a series of greater-than symbols to indicate the problem to us. If we enter

/GC4 RETURN

we can only obtain through YEAR 8 on the screen. Let's enter

/GC6 RETURN

to set the column width to 6 and examine another way to solve this problem.

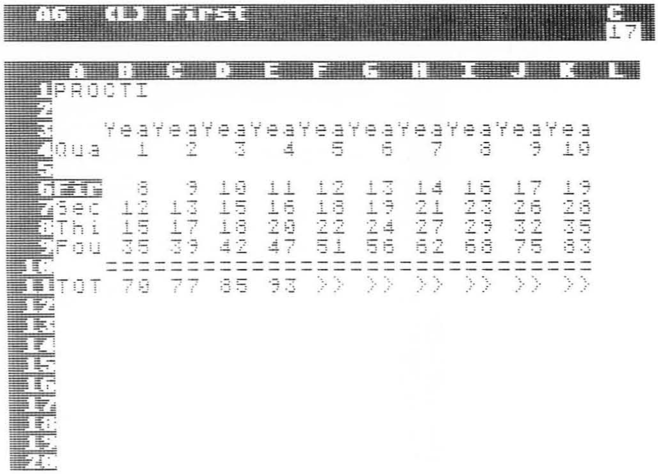


Figure 3-20. All columns of our report show on this screen when we reduce the column width to 3, but we no longer have room for the sum at the bottom of every column.

VisiCalc provides a split screen, which can be helpful in problems like this. Move the cursor to location A12 and type

/WH

which produces the screen of Figure 3-21. The functioning of the keystrokes here is explained next.

Keystrokes	Purpose
/W	Enter the Window command.
H	Establish a horizontally split window.

Now press the ; key. Press it again and again. This key jumps the cursor back and forth between the two windows. With the cursor in the bottom screen, use the arrow keys to move YEAR 6 through YEAR 10 into view as in Figure 3-22.

It's important to recognize that we are working with a limitation of a VisiCalc system. The size of the electronic sheet is often larger than the capability of the hardware to display on the screen. In our forecast, if we had a larger report, for example, a 20-year forecast, we simply could not see it all on the screen at one time. Even though VisiCalc provides powerful screen formatting capabilities, we are nevertheless often limited by the hardware.

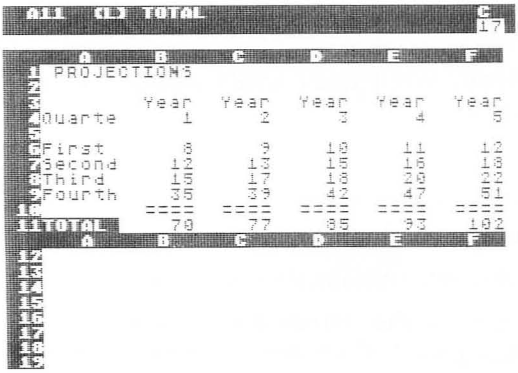


Figure 3-21. The screen has been split horizontally into two windows.

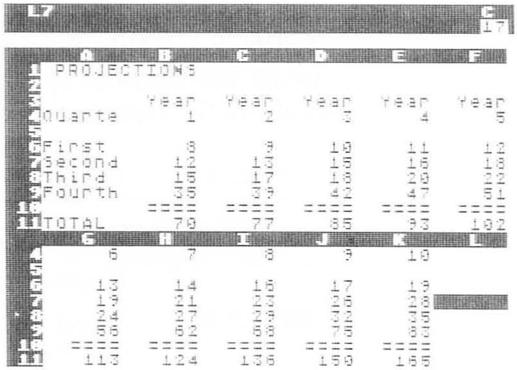


Figure 3-22. After scrolling in the lower window, we have our full forecast on the screen at once.

Use the ; key to jump to the top screen and then position the cursor to location B9. Remember that we started by forecasting for the next quarter, the FOURTH quarter of YEAR 1. We then added the additional years. Let's return to the FOURTH quarter and change that value. Enter the value 40. As we do so, we can watch a rippling down each column as VisiCalc updates each FOURTH quarter value (each is dependent on the previous one). Also notice that the character ! at the top right of the screen appears while the recalculation occurs. Change the value and watch this recur.

THE STORAGE COMMAND

We've completed a significant amount of work and want to retain this sheet. To do so, we'll introduce and use the Storage command. Chapter 4, Commands, contains additional information about this whole process.

Saving an electronic sheet requires either a blank diskette (not the VisiCalc diskette) or a cassette. For our example we'll work with a diskette. To initialize the diskette (a process required before recording data on a new diskette, or to cause existing unneeded data on a used diskette to be destroyed) we'll first place a new diskette into our disk drive and then type

/SI

We should then double check to be certain we have the correct diskette in the disk drive and when we're satisfied, press the RETURN key. This begins a process that requires a minute or so while the diskette is initialized. We'll relax while waiting for this and briefly explain the keystrokes

Keystrokes	Purpose
/S	Enter the Storage command.
I	Request the diskette initialization option.
RETURN	Begin and complete the initialization process. We'll see in Chapter 4, Commands, the importance of this confirming action.

We are now ready to save our sales forecast sheet on the recently initialized diskette. We type

/SS FORECAST RETURN

meaning

Keystrokes	Purpose
/S	Enter the Storage command.
S	Use the Save option of this command. At this point the box appears on the edit line, as VisiCalc waits for us to enter a file name.
FORECAST	This is a name selected for the sheet that we'll store on the diskette. Since we may store many sheets on one diskette, we need to name each one separately.
RETURN	Begin and complete the process of copying our sheet from memory to the diskette.

We have successfully stored our sheet on a diskette ready to be copied into memory when we need it again, perhaps after the FOURTH quarter, when we can enter the actual UNIT SALES.

Let's summarize all that we have done.

SUMMARY

Beginning with VisiCalc correctly loaded into the memory of our computer, we have gone through a series of steps to create an electronic sheet that forecasts unit sales over a 10 year period. In doing so, we have introduced many actions, all of which are explained in depth in subsequent chapters.

We have seen that VisiCalc provides an electronic two-dimensional worksheet. We have learned how to scroll the sheet, how to move the cursor, and how to write labels, numbers, and formulas on the sheet.

We have used formulas in a number of entries introduced and demonstrated one of the built-in functions available. Built-in functions can be a powerful tool in simplifying our work and reducing the possiblity of error.

A number of commands were demonstrated, including the Format, Repeating Label, Blank, Insert, Global, Replicate, Title, Window, and Storage commands. One of them, the Replicate command, provides a powerful tool that we used in several instances to generate a large number of relationships with a limited amount of typing on our part.

In subsequent chapters all of these, in addition to the remaining commands and built-in functions, will be explained.

Chapter 4

Commands

INTRODUCTION

This chapter contains a detailed explanation of each of the commands of VisiCalc. The commands are accessed by entering the slash (/) followed by the letter indicating the desired command. When a slash is entered, the following appears on the prompt line:

COMMAND: BCDFGIMPRSTVW-

Entering any of the characters after the colon initiates the desired command. The available commands are

B	Blank	P	Print
C	Clear	R	Replicate
D	Delete	S	Storage
F	Format	T	Title
G	Global	V	Version
I	Insert	W	Window
M	Move	-	Repeating label
		X	Unnamed

This chapter contains a detailed explanation of each of these commands.

/B THE BLANK COMMAND

The Blank command, /B, is used to blank, or clear, a single entry on the sheet. The action occurs when we type

/B

followed by an arrow key or the RETURN key. To cancel the action (after typing /B), type the BREAK key. In fact, many other keys will also cancel this command, but to be certain let's establish the BREAK key as the cancel key, its intended purpose. The large, easy-to-locate space bar will also cancel this and many other commands. Canceling the Blank command leaves the contents of the entry unchanged.

As an example of the action of this command, compare Figure 4-1 with Figure 4-2. We want to blank entries A17 and A18, starting with A18 where the cursor is located. After typing

/B RETURN

the entry is cleared, as we see in Figure 4-2.

Let's move the cursor to location A17 and note on the entry contents line that the entry has a format, /FL (left justification), and a value of 517 as shown in Figure 4-3. Notice in Figure 4-4 that the format remains after executing the Blank command. Blank does not affect the format of the entry. To remove the format, use /FD as described later in this chapter.

The command, when executed, does not become the entry of

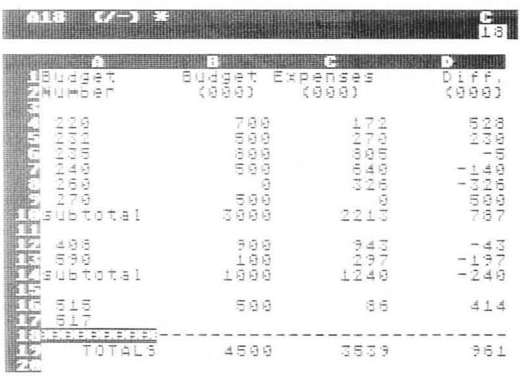


Figure 4-1. A sheet on which we want to blank entries A17 and A18. The cursor is located at entry A18.

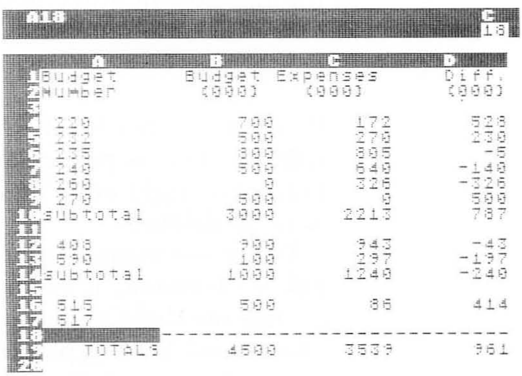


Figure 4-2. The same screen after the Blank command is used.

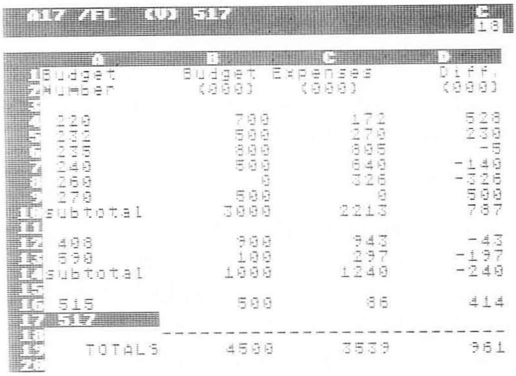


Figure 4-3. We want to blank the format and value of entry A17 (note the entry contents line).

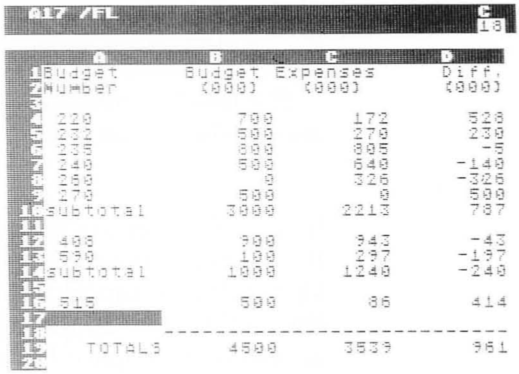


Figure 4-4. The same screen after the Blank command is used. Note that the format remains.

the location and does not itself require storage if the sheet is saved.

Blank entries are evaluated as zero (0) when they are referenced in formulas in other entries.

This command can be used in combination with the Replicate command to blank large areas of the sheet, as Figure 4-5 demonstrates. This sheet has a column of erroneous values (garbage) from C20 through C31, which may have been generated by incorrect use of the Replicate command (described later in this chapter). In Figure 4-5 we have blanked entry C20 (/B RETURN), removed the format (/FD), and begun the process of replicating the now blank entry down the column. We have left the cursor at entry C20 and entered

/R:C21.C31

When we press RETURN, the blank entry is replicated down the column. This is particularly effective on areas to be blanked on the bottom or right edge of the sheet, where multiple rows or columns may be deleted.

When we want simply to replace one entry with a new value, label, or formula, it is not necessary to use the Blank command as an intermediate step. Simply moving the cursor to the desired location and then typing the new entry will result in replacing the old contents with the new contents.

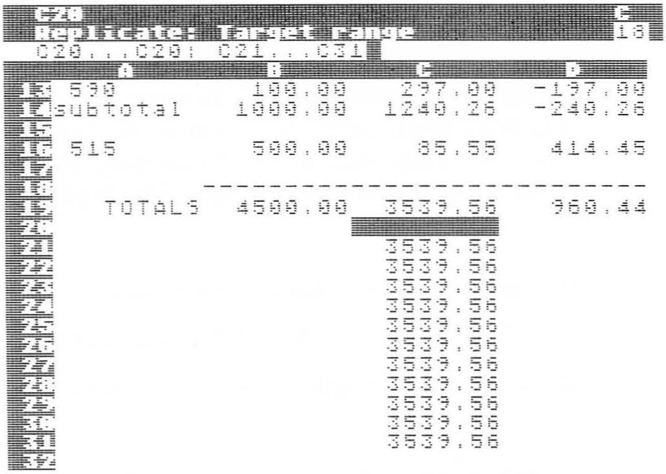


Figure 4-5. A spreadsheet on which we have blanked entry C20 and are preparing to blank entries C21 through C31, using the Replicate command as shown.

/C THE CLEAR COMMAND

The Clear command, /C, is used to clear the electronic sheet. Its action is final, and for this reason there is a built-in requirement that we verify our desire to destroy the existing contents of the sheet. The command is initiated by typing

/C

at which time the prompt line reads

CLEAR: Y TO CONFIRM

If we want to complete the action, type

Y

at this point. The screen is blank for several seconds and then reappears as if we were loading VisiCalc from the start.

To cancel the command, type BREAK or use the space bar. In fact, other keys will also cancel the action, but as discussed under the Blank command, let's establish the habit of using the BREAK as our cancel key. The reason for not typing "any" key is clear in this case: consider what happens if we enter /C, then decide that we do not want to destroy the sheet. Suppose we simply pressed any key, and in fact press the Y key. We will destroy the sheet.

Clearing the sheet means that

- All entries are set to blank.
- The global parameters are set to general:
  The column width is set to 9.
  The order of calculation is columnwise (explained later in this chapter).
  The recalculation mode is automatic (explained later in this chapter).
  The format is general, indicating no particular global format.
  The window is set to 1 (single window).
  The cursor is positioned to entry A1.
  There are no fixed titles.

Figures 4-6 and 4-7 illustrate this action. In Figure 4-6 we see the screen before the Clear command; and Figure 4-7 shows the screen after that command. After the Y is typed, the screen appears as in Figure 4-7.

Clearing affects the sheet currently stored in memory but does not affect this or any other sheet stored on a diskette or tape. If we clear a sheet in memory that we want to restore, we can do so

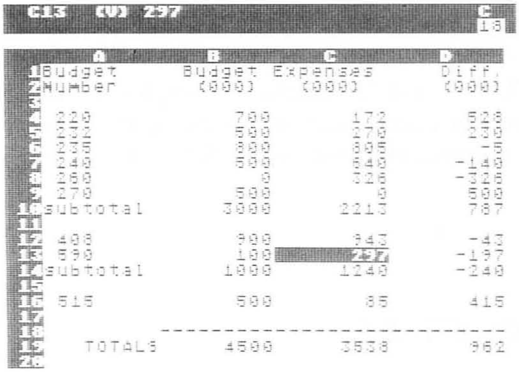


Figure 4-6. An electronic sheet before clearing.

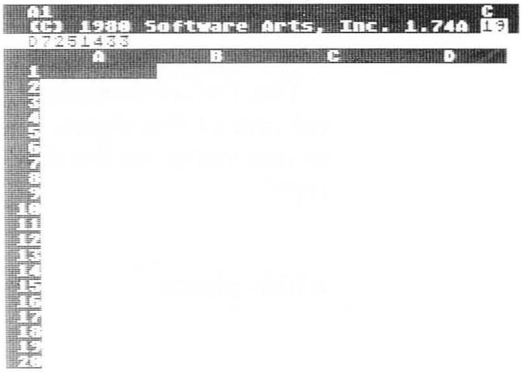


Figure 4-7. Our screen after confirming, by typing Y, that we want to clear the sheet.

only if we have saved it. This emphasizes the importance of creating backup copies of valued spreadsheets. We'll discuss this again in this chapter in the Storage command section.

The Clear command should be used between finishing work on one sheet and before starting on another at the same sitting. If the first sheet is to be saved, do so before issuing the Clear command. Then clear the sheet. If a second sheet is to be loaded from diskette or tape, do so after clearing memory with this command. If one sheet is loaded on top of the other without issuing the Clear command, an incorrect combination of the two sheets may result. (Using this action constructively as an overlay technique is considered under the Storage command.)

We should never think of this command as a continuous three keystroke /CY command. Instead, it's important to consider this command as a three-step process:

Step 1. Type /C.
Step 2. Hesitate to be certain that we want to clear the sheet.
Step 3. Type Y.

This will ensure that we use the confirmation action of this command appropriately.

/D THE DELETE COMMAND

The Delete command, /D, is used to delete a complete row or column of the sheet. The action occurs when we move the cursor to any entry on the row or column that we want deleted and then type

/D

which places

DELETE: R C

on the prompt line. We then enter

R to delete the row
or C to delete the column.

To cancel the delete action, we'll type BREAK or use the space bar instead of an R or C. Other keys will also cancel the action, but as in discussions of /B and /C earlier in this chapter, let's use the BREAK or the space bar as our cancel key.

No confirming action is required although it would have been desirable to require a RETURN or other action to help prevent inadvertant deletion. There is no restoring the row or column after deletion. Therefore, caution is necessary, and the need for backup of spreadsheets, emphasized in many other places of the book, is clear. A rippling occurs through the sheet as it is rewritten without the deleted row or column. With a large sheet this will require several seconds with no indication that anything is occurring.

All rows below the deleted row (or to the right of the deleted column) are moved up one row (or one column to the left).

The action of deleting either a row or column will usually affect formulas on the sheet. Look at Figure 4-8 and notice that the formula on the entry contents line for location B19 is

(B10+B14+B16)

Now let's delete row 3. We'll move the cursor up to B3, as shown in Figure 4-9 (any entry on row 3 would be suitable), type

/D

look at the screen to verify that we've located the cursor correctly, and then type

R

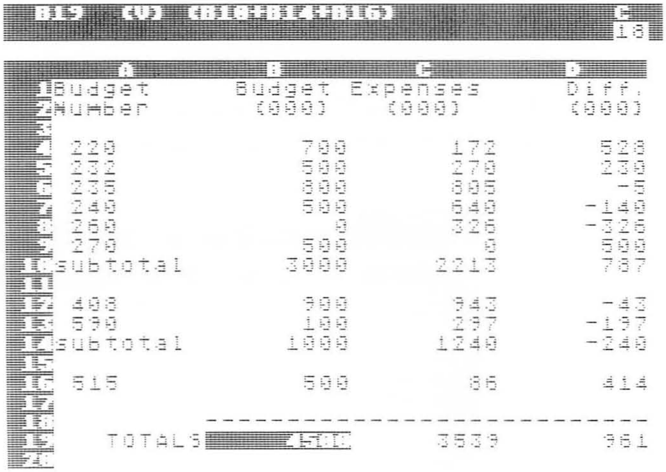


Figure 4-8. One entry of the sheet (B19) before deleting a row. Notice that the formula here is (B10+B14+B16).

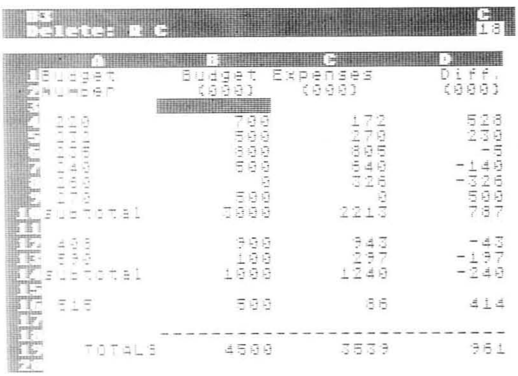


Figure 4-9. Preparing to delete row 3 of this sheet.

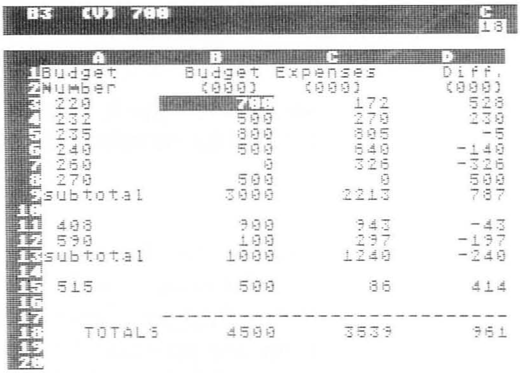


Figure 4-10. After deleting row 3 of Figure 4-9, we have this new sheet with a new row 3.

This hesitation after typing /D and before typing either R or C is important because it can prevent the unwanted deletion of a row or column. Notice in Figure 4-10 that all entries below the old row 3 have moved up one row and our cursor is still in position B3 (which now has value 700 in it). Now let's look again at the total for this column, in Figure 4-11, by moving the cursor to location B18.

Now the contents of this entry are

(B9+B13+B15) not (B10+B14+B16)

and we see that VisiCalc has adjusted this formula, and all others, to reflect that a row has been deleted.

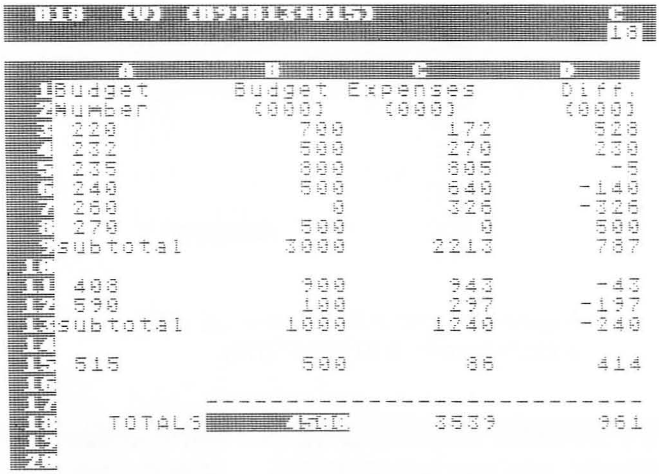


Figure 4-11. The formula in the total entry of column B is revised by VisiCalc to adjust for the deletion of the third row. Compare this to Figure 4-8.

Let's consider what happens when we delete a row in the middle of a series, such as the row containing BUDGET NUMBER 235 of our budget. Refer to Figure 4-12 as our starting point. The formula for position B10 indicates that it is evaluated as the sum of entries B4 through B9. That sum is currently 3000, which includes B6, with a value of 800, on the row that we want to delete. Let's delete row 6, and then return the cursor to the location of the sum that we're discussing. This is now location B9 as shown in Figure 4-13. Notice that our new formula sums B4 to B8, the new range, and that the total is no longer 3000 but 2200. Row 6 is gone and its value is no longer part of the sum.

There are times when deleting a row or column can cause problems. Let's delete row 8 from Figure 4-13, and look at the results shown in Figure 4-14.

Notice that six entries show ERROR and that the entry at location B8 shows a formula of

@SUM(B4...@ERROR)

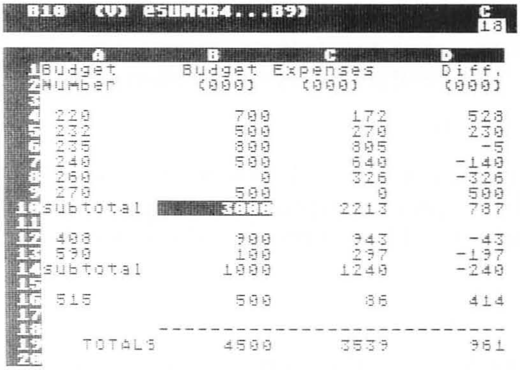


Figure 4-12. One entry of the sheet containing a series reference from which we'll delete a middle row.

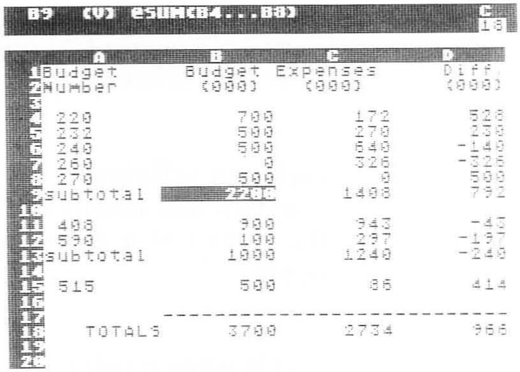


Figure 4-13. Our formula, after deleting a row from the middle, has had the closing entry of the series altered to compensate for the missing row.

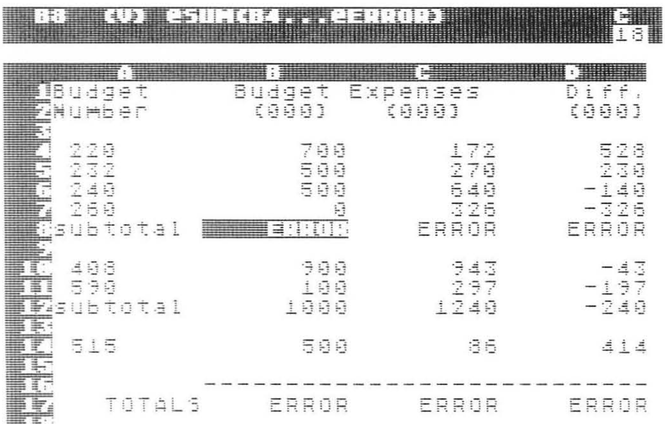


Figure 4-14. If we delete the row with the last element of the series, we cause ERROR entries in the sheet.

VisiCalc has not replaced the old B8 in the sum with another coordinate but has instead replaced it with the @ERROR function, indicating that no computation will be performed. (The @ERROR function is discussed in Chapter 6, Built-in Functions.) Location B17, which references B8, is also ERROR, and similar results have occurred in the other entries of rows 8 and 17 of Figure 4-14. This problem is discussed again, and a solution suggested in the discussion of the @SUM function.

As with other commands, the hesitation between entering

/D

and then selecting R or C is very important. For many users, the insertion and deletion of blank rows to improve the appearance of the sheet are a common activity. Thus, entering the three keystrokes

/DR

may become a habit. We should avoid this, since we may inadvertently delete a row when we really wanted to delete a column. We can prevent this by developing a hesitation after entering /D with a conscious decision before entering an R, C, or BREAK (to cancel).

/F THE FORMAT COMMAND

The Format command, /F, is used to establish a format for a single entry of the sheet. It is initiated by typing

/F

at which time the prompt line reads

FORMAT: D G I L R $ *

The available formats are

D	Default
G	General
I	Integer
L	Left justified
R	Right justified
$	Dollars and cents
*	Graph

Format indicates the method that VisiCalc will use to display a value or label on the screen, or in a printed report when part or all of an electronic sheet is printed. This command affects only the single entry in which it is entered, although we'll see that it is possible to use the Replicate command in conjunction with it to simplify reproducing various formats on the sheet. The format becomes part of the entry and is displayed on the entry contents line (except for the D format).

It is important to realize that this format affects the appearance of the entry and not the method of storage. For example, a number may have a decimal part that will not be displayed with the integer format; however, the integer format only affects the display. The decimal part of the number remains stored, and the full value is used in any references in formulas.

Each of the format options is discussed next.

D Default to Global Format

The D format indicates that this entry should be displayed with the global format. The entry contents line for a field with the D format indicates no format, meaning that this should default to the global format in effect. Since an entry with no format defaults to the global format, this particular entry is used only to erase an existing format within an entry or in conjunction with the Replicate command to remove formats from many entries (as discussed earlier in this chapter in the Blank command section).

G General Format

The G indicates the general format (not the global format obtained with the Default D option).

For labels, the specification acts as the left justification (L) format; a field containing a label and the G format remains left justified even if the global format is right justification.

For numbers, this format will display values in either normal (as entered or computed) format or in scientific notation, depending on the column width. Numbers that are entered with leading or trailing zeros will be stored without them. Leading and trailing zeros and scientific notation all are discussed in Chapter 5, Labels, Numbers, and Formulas.

I Integer Format

This format indicates that the field will be presented as an integer value, although as with other values it may be stored as a noninteger. As with other formats this affects the way in which the number is presented, not the stored value. If the entry contains a number that is not an integer or a formula whose current value is not an integer, the value will be rounded and displayed as an integer. Rounding means that if the decimal portion of a positive value is greater than or equal to 0.5, then the next larger integer is displayed; otherwise the next smaller integer is displayed. With a negative value, if the decimal portion is greater than or equal to 0.5, then the next smaller integer is displayed; otherwise the next larger integer is displayed.

For example, four values are shown as they are stored and as they will be displayed with an integer format.

Stored Value	Displayed Value, /FI
6.8	7
6.3	6
−6.3	−6
−6.8	−7

L Left Justified Format

For VisiCalc, this format means that all characters of a *label* entry will appear, starting in the leftmost position of the field, and that all characters of a *value* displayed will appear, starting in the second leftmost position of the field with a blank as the first character. For numbers, this blank prevents several fields from being run together when displayed. If we want to enter numeric characters that are fully left justified, they must be entered as a label, for example, by entering

"50

in a field. When this is done, the characters 50 are a label and not a value, meaning that if it is referenced in a formula or other calculation it will be evaluated as 0 (zero) and not as the value 50.

With VisiCalc, all labels are left justified by default, so that the /FL format is redundant when used with a label unless the global format is right justified.

R Right Justified Format

This format means that all characters of a label entry will appear with the last character ending in the rightmost entry of the field. For labels larger than the current column width, this format is ignored; that is, the entry is filled with characters starting with the leftmost of the entry.

Values are right justified by default so that this format is unnecessary for values or formulas unless a global left justification is in effect.

$ Dollar and Cents Format

This format displays numeric values with two decimal places. Entries evaluated with more than two decimal places will be rounded to two places. Entries evaluated with less than two decimal places will be shown with two places; for example, 16.5 will be shown as 16.50. Dollar signs ($) and commas (,) do not appear. This may be undesirable, depending upon the application. A $ may be right justified in the field immediately to the left of the value in order to give the desired appearance for some applications. If appearance is significant and the entry will not be used in computation, then it can be entered as a label. For example, a price list could include an item description followed by the price included both as a label and as a value, such as

Description:	Price (L)	Price (V):
MICROCOMPUTER	$2,867.00	2867.00

In this example the middle column is a label that could be used in printing reports, while the right column is a value that could be used in computation where the price is required. The middle column is right justified (/FR) and entered as a label

"$2,867.00

Care is of course necessary to ensure that the two are the same and that if the price changes, both columns are changed.

* Graph Format

This format provides a simple bar graph capability. It left justifies asterisks in the field (starting at the second position of the entry). The number of asterisks placed is equal to the unrounded (truncated) integer portion of the value at the entry (with the exceptions that follow). When truncated both 9.1 and 9.8 become 9.

The exceptions are

- If a label is placed into a field with a /F* format, the graph format is ignored (although not destroyed) and the label is left justified.
- Values less than one, including negative values, or formulas evaluated in this range are displayed as a blank field.
- A value or formula whose truncated integer portion is equal to or greater than the width of the field will be displayed as the field filled with asterisks except for the blank in the leftmost character. Thus, a field of width 9 whose current value is 9 or greater (9.1, 10, 15.37, etc.), will be displayed as a space followed by eight asterisks. No error indication is given when this occurs, which is undesirable for many applications.

Considerations in Formatting

There are a number of important considerations in using the Format command.

- Every entry has a format even if one is not specified with this command. If no explicit format has been entered with the Format command, then the entry has the global format in effect at the time.
- Any rounding of values (for example, with /FI or /F$) affects only the display of the value and does not affect the way in which the number is stored or evaluated on the sheet.
- Only one explicit format can apply to a single entry. It is not possible to have a single entry in dollars and cents format (/F$) and also left justified (/FL). However, dollars and cents and right justified do occur together since right justified is the implicit default justification for numeric values. Similarly the integer built-in function (@INT) could be used to provide an integer format in conjunction with an explicit format specification, for example left justification.
- This command is local, meaning that it only affects the one entry at the cursor location where it is entered. However, we'll see that the F can be used with the Global command to establish formats for all entries without a local format and can also be used with the Replicate command to repeat formats in a number of entries.
- The formats established with this command take precedence over the global format setting. That is, changes in the global format (see the next section of this chapter) will not affect these individual entry settings.
- The column width significantly affects the display of values and labels; therefore, a combination of formats and column width determines how an item will be displayed. This will be discussed in Chapter 5, Labels, Numbers, and Formulas.
- It is possible to use the Replicate command (/R) to replicate formats in the same way that other items are replicated. If there is a value, label, or formula in the entry or entries being replicated, then both will be replicated. Where this is undesirable, then either the format or the values, labels, or formulas must be entered individually.
- There is no way to erase all local formats with one command.

Also, no replicate-format command allows for replicating only the format of an entry, and not its value. However, the format can be entered first and replicated, and then values entered individually later.

• It is possible to generate formats in areas of the sheet that we expect to use, and then to leave them there in error when we do not use that part of the sheet. Similarly, it's possible to use the Blank command (/B) to blank values, labels, or formulas and unintentionally leave formats intact. If it's desirable to remove these formats, then the default option of the Format command should be used (/FD) as already described in this section.

• The formatting process in many data processing applications often requires more effort than the procedural aspects of the problem solving. This is true with many VisiCalc applications where correct placement of labels, setting of column widths, and formatting of individual entries may require significant activity on our part.

/G THE GLOBAL COMMAND

The Global command, /G, is used to control, for the entire sheet, column width, order of recalculation, mode of recalculation, and format. It is initiated by typing

/G

at which time the prompt line reads

GLOBAL: C O R F

These correspond with the available global commands, which are

C	Column width
O	Order of recalculation
R	Recalculation mode
F	Format

Each one of these is discussed next.

C Column Width

It is possible with this option to vary the width of the columns appearing on the screen from three characters to thirty-seven. To change the width, enter

/GC

at which time

COLUMN WIDTH

will appear on the prompt line. You should now enter the desired number. After the first digit has been entered, a box appears and the software is waiting for us to enter a second digit or a RETURN, as appropriate.

The column width can be different in two windows of the screen. Figure 4-15 demonstrates this and also demonstrates the maximum and minimum allowable column widths for this version of VisiCalc. The top screen has a label in entry A2; only one column, here column A, fits on the screen. Notice that it extends fully to the rightmost edge of the screen. In the lower window a global column width of 3 has been entered. Notice that 12 columns (A through L) appear and that the total width of all columns is one less than the number on the top screen.

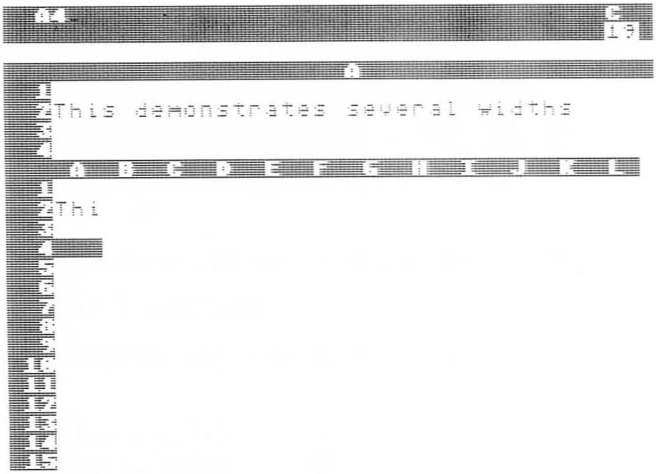


Figure 4-15. A split screen showing the maximum column width of 37 for this computer in the top window and the minimum column width of 3 in the lower window.

Also notice that the label at location A2 in the lower screen contains

THI

and not the full message stored at A2. It has been truncated (cut off) from the right, leaving only the leftmost three characters. The full label, as stored on the sheet, is unchanged; only its presentation on the window is altered.

Printing is affected directly by the column width, and reports prepared from the sheet will have the column width of the screen. If there are two windows with different screens, the column width of the window from which the report is printed will be used. (Refer to the Print command.)

The global column width is also discussed in Chapter 5, Labels, Numbers, and Formulas.

O Order of Recalculation

A recalculation of all entries of the sheet occurs, either by changing an entry or by entering an exclamation point (!) to initiate a recalculation. When we do so, the software can either perform the recalculation in column order, C, as indicated by the arrows in Figure 4-16, or in row order, R, as shown by the arrows in

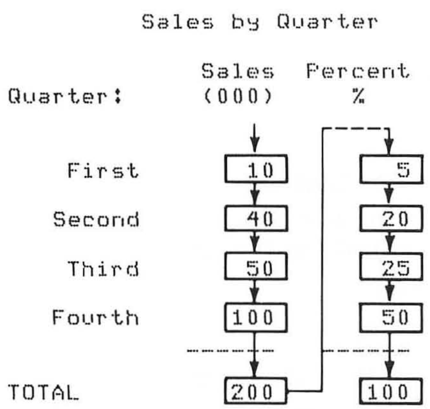


Figure 4-16. A sheet graphically indicating the meaning of columnwise recalculation order, C.

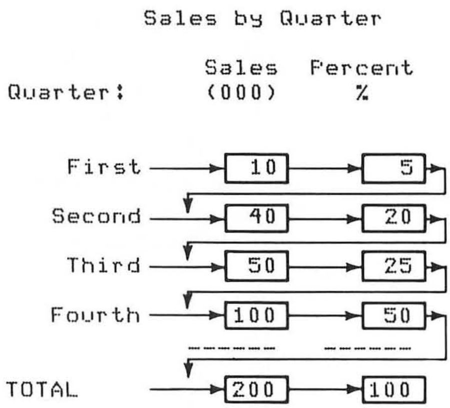


Figure 4-17. A sheet indicating rowwise recalculation order, R.

Figure 4-17. In both cases the action begins at entry A1. These figures contain a simple sales-by-quarter report in which quarterly sales are totaled in the SALES column and then each of the four sales entries is divided by the total to determine the PERCENT for that quarter.

By default the recalculation order starts at entry A1 and proceeds down the A column, then continues at B1 and moves down the B column, etc. However, if we want to go across, we enter R, as described below, and we will start at A1 and go across row 1, return to A2, and then go across row 2, etc.

In the upper righthand corner of the screen, on the entry contents line, is the recalculation order indicator as shown in Figure 4-18. This is either a C or an R, depending on our action. When we enter

/GO

the prompt line indicates

REEVAL ORDER: R C

We now enter either R or C, indicating the desired reevaluation order. They mean

R	Rowwise recalculation
C	Columnwise recalculation

Let's look more closely at the last examples to demonstrate the difference between these two. Suppose that we begin with the example of Figure 4-17 with an R (rowwise) recalculation order. Look at Figure 4-18. It has been derived from Figure 4-17 by changing the entry for the FOURTH quarter from 100 to 0. Note that the SALES column now correctly totals 100. However, look at the PERCENT column. The individual percentages are all incorrect. The PERCENT total is 50% and not 100%. The formula for the percentage entry on the FIRST row has been computed by dividing FIRST by TOTAL and multiplying the result by 100, that is,

(B7/B12)*100

This is the correct formula but the wrong answer. The problem is that when this was computed, the value at B12 was still 200 and not the total of 100, which was correctly placed there later.

Let's follow a simple trace through the reevaluation to emphasize how this occurred. To do so, we begin with the values shown in Figure 4-17. The trace is a step-by-step tracking of what occurs. Let's follow the action until we discover the cause:

Entry Location	Entry Contents	Evaluation	Notes
B10	100	100	We position the cursor to this location in order to change our FOURTH quarter value from 100 to 0.
B10	0	0	We make the change, triggering the recalculation of the entire sheet, which occurs after the value here is changed.
			The reevaluation occurs, sweeping across the rows, from A1 to B1 to C1, then A2, B2, C2, etc. Nothing is in fact changed until the formula at C7 is encountered.
C7	(B7/<u>B12</u>)*100	(10/<u>200</u>)*100 5	Here is the problem. Entry <u>B12</u> still has the old total of <u>200</u>. It is not updated until the other rows have been reevaluated. As a result, each percentage is wrong.

At this point we see the value of the exclamation point. If we

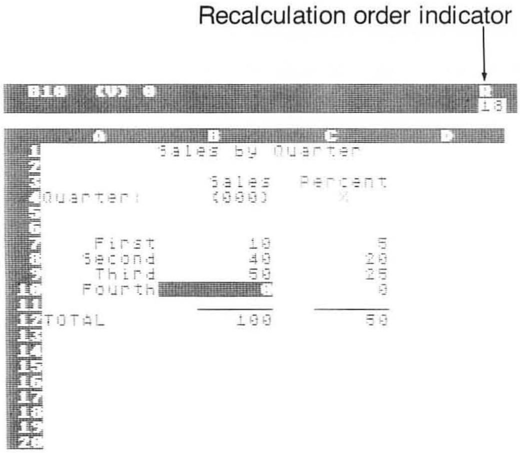


Figure 4-18. The rowwise order of recalculation has resulted in this incorrect sheet. Note that the percentages are all incorrect even though the formulas are correct.

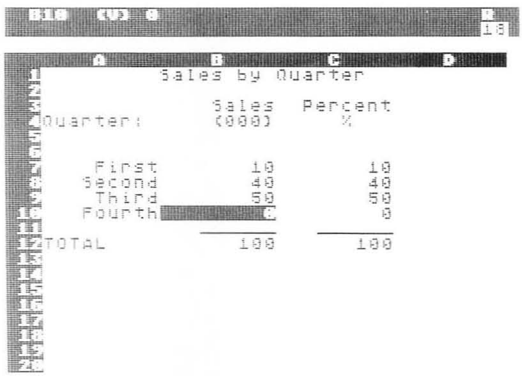


Figure 4-19. The sheet of Figure 4-18, now correct, after a recalculation initiated by entering an exclamation point (!).

enter !, our sheet will be fully recalculated, providing the correct values as shown in Figure 4-19. We get correct results because the value at entry B12 is 100, and no longer the older value of 200.

Why use rowwise recalculation? For this sheet it was incorrect; however, its use can be demonstrated by a different presentation format of the same data. Suppose that we wanted to prepare this report in the format shown in Figure 4-20. For this report the correct order of reevaluation is rowwise. This is true because we must compute the value at location F8 before we compute any value in row 9.

With this understanding of the differences between row and column calculation order we can emphasize the need to develop models which do calculate correctly. We want to build our sheets so that each entry is a function of entries to its left and above its location. There is additional discussion of this concept in Chapter 7, Other Topics, under the discussion of Forward Reference.

A problem in the order of recalculation can become apparent when a sheet is stored and later reloaded into memory. If a value is required before it has been placed on the sheet, then ERROR is

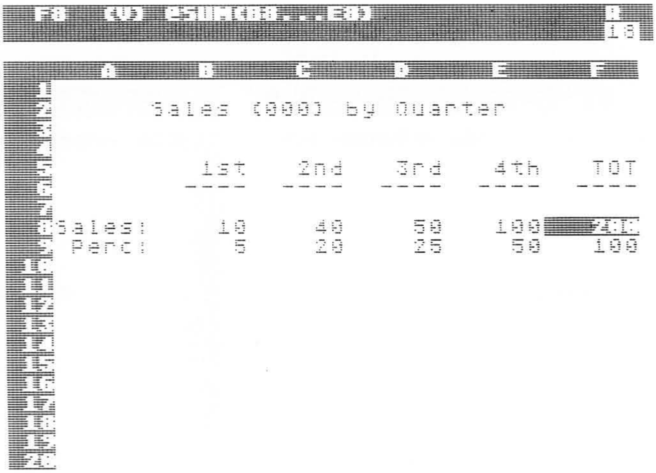


Figure 4-20. The same problem formatted in a way in which the correct order of recalculation is rowwise.

presented. These will disappear when a recalculation (!) occurs.

Let's summarize:

- VisiCalc defaults to a columnwise order of recalculation.
- Users can change the recalculation order by use of the O option of the Global command.
- The correct choice (C or R) will depend on the requirements and layout of our solution to the problem.
- Entering an ! will usually eliminate any problems and is suggested before using the results.

R Recalculation Mode

On large sheets, or on small sheets with lengthy or time-consuming formulas, the recalculation that automatically occurs when any value on the sheet is changed can take several seconds, or longer, to complete. Although not long, even this delay can be frustrating, especially when many changes are being made on the sheet. VisiCalc defaults to this automatic, instantaneous recalculation mode; however, it provides the capacity to shift to a mode in which a recalculation occurs only when we want it.

This is accomplished by entering

/GR

which generates

RECALC: A M

on the prompt line. The two recalculation modes are

A Automatic
M Manual

Automatic is the default and produces a recalculation wherever a value or formula is entered or revised. Manual postpones this recalculation until we initiate it by entering an exclamation point. Figure 4-21 contains a sheet on which the recalculation mode has been set to M, for manual. Notice that the TOT is shown as 0 (zero). At this point, assume that we want to see the total, which we can obtain by entering !. It can also be obtained by changing the mode to automatic, which activates a recalculation. Figure 4-22 shows the result of entering the !. While a recalculation is occurring, an ! appears in the upper right of the screen until the action is completed.

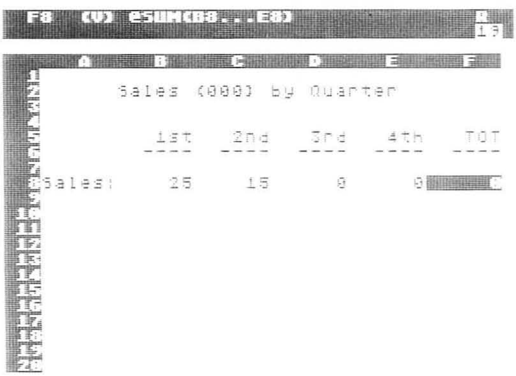


Figure 4-21. A spreadsheet on which the manual recalculation mode is in effect. Note that the total (TOT) column is temporarily incorrect.

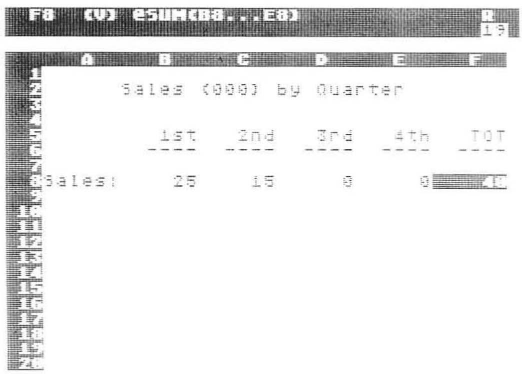


Figure 4-22. The sheet of Figure 4-21 after a recalculation (!). The total (TOT) is computed and correctly displayed.

Since a manual recalculation mode can temporarily leave the sheet with incorrect values and since there is no visual indicator of which mode is in effect, it's important to depress the ! before relying on values on the sheet.

F Format

All of the actions of this option are similar to those listed in the last section on the Format command, but the actions here affect all entries of the sheet that do not have an individual format. In this usage, the format entered becomes the default format. It does not appear on the entry contents line as the Format command does.

If a split screen is being used, the global format only applies to the half in which it was entered.

To return to the original global format, the /GFG (global format general) is used.

/I THE INSERT COMMAND

The Insert command, /I, is used to insert a complete row or column on the sheet. The action is initiated by typing

/I

which produces a prompt line of

INSERT: R C

At this point, entering an R causes a row to be added above the current cursor location, and entering a C causes a column to be added to the left of the column in which the cursor is located. The new row is inserted above the cursor row (rather than below it); otherwise it would be impossible to add a new row above row 1. The same action is true with adding a column to the left of the cursor.

To cancel the action, depress the BREAK or the space bar.

No confirming action is required. The sheet ripples as the insertion occurs, which on a large sheet may require several seconds.

This command inserts a blank row or column. If we wish to have formulas, values, or labels in place on the newly inserted row or column, we can do so by writing on the newly expanded sheet.

As with the Delete command, formulas are automatically revised as necessary to accommodate the new row or column. When a row or column is added to the middle of a series, for example, with the @SUM function, the values placed in the row or column will be included in the series. A review of the description for the Delete command will show what occurs. When a row or column is added at the start or end of a series—for example, a series that is an argument of the @SUM function—the series is not revised to include the new values. The discussion of the @SUM function in Chapter 6, Built-in Functions, presents one way of including this new row or column when it appears at one end of the series. The description of the Delete command earlier in this chapter also includes a discussion of the problems associated with deleting in series references.

Although it is sometimes desirable to have the capability to add more than a single row or column, only one row or column can be added at a time.

As suggested in Chapter 9, Creating Templates, blank rows or columns included for appearance should be inserted as the last step in developing a sheet. This will prevent undesired entries in the blank row or column, which can be generated when replicating down or across areas of the sheets that should be blank.

Although VisiCalc does not have a sorting capability, it is sometimes possible to perform a crude sort by inserting. If we have an unsorted list of items, for example, employee names, we can enter the names one after the other by inserting a blank row in the appropriate position and then entering the next name onto the blank row. Since this is being done "by eye," it is not a reliable sorting activity but can be helpful in some instances.

/M THE MOVE COMMAND

The Move command, /M, is used to move either a full row or a full column to another location on the sheet. It is initiated by typing

/M

which places

MOVE: FROM...TO

on the prompt line.

The same command is used to move either a column or a row; only a starting FROM coordinate and an ending TO coordinate are provided. Therefore, if we specify a move from entry B12 to B6, we are indicating a move of a row, that is, move row 12 to row 6. Similarly, if we move from E6 to B6, we are specifying a move of a column, here move column E to column B. If we specify a move from B6 to G12, no move occurs and no error is indicated. In the last case both the row and column identifiers are different and thus the command structure cannot determine whether this is a row or column move.

The Move command has an action similar to the action involved when we have a pile of 3×5 cards and move a card from one position to another. If we move one card forward in the deck, all cards behind its original position do not move, and all cards behind its new position back to its old position are bumped down one relative position. It is a real move, which means that the card no longer exists at its old location.

Similarly, if we move a card back in the deck, cards are shifted forward as the card is removed and replaced in its new position. In VisiCalc a forward move—that is, down for a row or to the right for a column—leaves the TO column (or row) intact and places the moved column (or row) to the left (or above) the TO location.

Let's look at the example in Figure 4-23, in which we have determined that we want to move the row with sales by NOAH to between MARTHA and TOMMY, its correct alphabetic position. We position the cursor to A5 (any entry on row 5 would be satisfactory) and enter

/M

which is what has occurred in Figure 4-23. At this point we have A5 followed by a box. We can

• Press BACK S to erase the A5 and reposition the FROM entry if we realize that we have an error.

• Enter BREAK to back out of this command if we want to cancel the action (the space bar will not work here).

• Enter a period to include an ellipsis at this point in our series, then type our desired TO location.

• Use the arrow keys to move the cursor to our desired TO location. This action automatically includes the ellipsis in the middle of the series.

Suppose that we follow the last listed action, moving the cursor from A5 to A8, leaving

A5...A8

on the edit line followed by a box. If we complete the action by entering the RETURN key, we'll have moved the row to its desired position as shown in Figure 4-24. The cursor remains at A5, the FROM location.

Notice that all subtotals and final totals are unchanged and still correct.

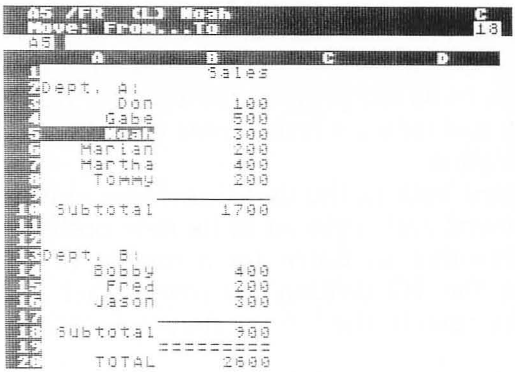


Figure 4-23. A screen in the middle of the process of moving the row with NOAH to its correct alphabetical location in DEPT. A.

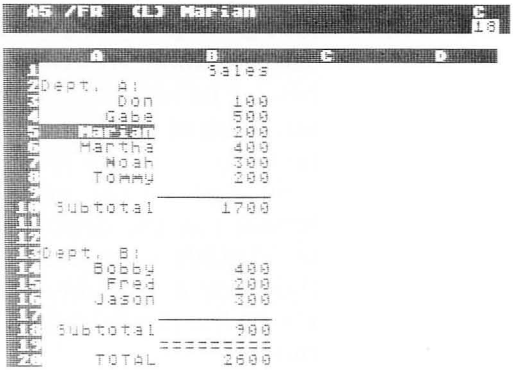


Figure 4-24. The screen after completing the move described in Figure 4-23.

Now suppose that we realize that TOMMY is actually in DEPT. B and that we want to move him to that department and list him after JASON. Figure 4-25 shows the screen after completing this move.

Notice the two subtotals, both incorrect. The formula for the SUBTOTAL of department A shows

@SUM(B3...B16)

at location B9 and for department B shows

@SUM(B13...B15)

at location B18.

The formula at location B9 has been revised by VisiCalc to show the new location of TOMMY, now B16; the formula at B18 has not been revised to contain a new end entry for the series. In fact, repeated recalculation (!!...!!) will continually change the values. This is caused by the "forward reference," which is discussed in Chapter 7, Other Topics.

We'll need to use the Move command with care because of the type of problems already described. There is additional consideration of this problem and a partial solution suggested for the series problem in the discussion of the @SUM function.

Moving only part of a row or column, instead of a complete row or column, cannot be accomplished with this command, although that can be done in several steps by using the Replicate command and the Blank command.

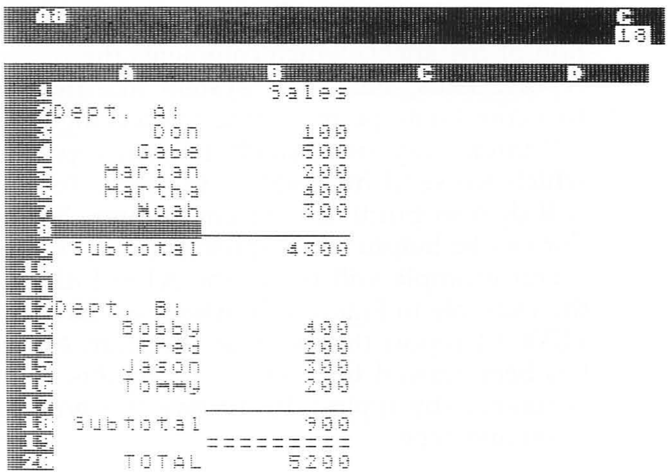


Figure 4-25. Errors introduced by moving TOMMY from the last position of DEPT. A to the last position of DEPT. B. Neither SUBTOTAL is now correct.

/P THE PRINT COMMAND

The Print command, /P, is used to print all or part of an electronic sheet or to write a file to a diskette or other device containing the current labels and values of the entries (rather than the formulas). It is initiated by typing

/P

at which time the prompt line reads

PRINT: PRINTER, FILE

which prompts for one of two items. They are

P (for PRINTER), meaning print to paper.

F (for FILE), meaning write the file to a diskette with only the labels and values. This creates a file that can be accessed by programs other than VisiCalc. It can also be used to print to a serial printer.

The Print command is conceptually straightforward. We begin with information on an electronic sheet that we want to print. We will consider all printed reports as a rectangle. To print a report, we place the cursor at the top left corner of the portion of the sheet that we want to print, and we issue the Print command. The command includes pointing to the bottom right corner of our portion of the sheet, which completes the rectangle and produces our report. Thus, on a grid system like the worksheets of VisiCalc, two coordinate points define the rectangle that we want to print.

Printers vary significantly in their capabilities and in the way in which we send information to them. To simplify this section, we will discuss printing with one printer. For other printers our vendor can be helpful in supplying specific information.

Our example will be for the ATARI 825™ printer. Let's look at the example in Figure 4-26 where we have on the screen a SIMPLE BUDGET report that we want to print. Notice that the cursor here has been moved to position A1. At this point we initiate the Print command (by typing /P), receive the prompt line previously indicated and type

P

which places

PRINT: LOWER RIGHT, "SETUP, +, −, &

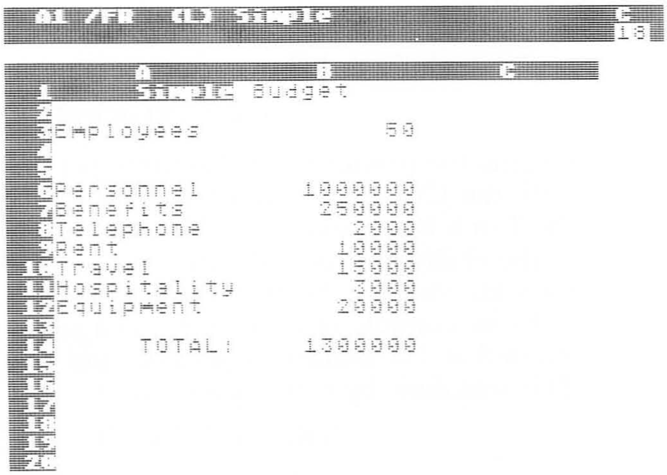


Figure 4-26. A screen displaying an electronic sheet that we want to print.

on the prompt line. This somewhat confusing prompt line is requesting one or more of several possible actions on our part. They are

LOWER RIGHT, meaning type the coordinate of the bottom right (LOWER RIGHT) of the rectangle that we want printed, here B14, or use the arrow keys to move the cursor to the desired entry. When followed by a RETURN, this begins the printing, giving us the printed report of Figure 4-27. Notice that the report is single spaced on this printer. This prints the report at 80 characters per line.

The lower right coordinate can be anywhere on the sheet and does not need to be currently visible on the screen.

''SETUP, meaning the option to choose for a number of other actions, for example to print 132 characters per line. Here we are requested, or prompted, to enter the double quote key followed by what's called a SETUP string. SETUP strings allow us to send a series of characters to the printer to establish settings for, or control the carriage, line feed, and other features of individual printers.

The string does not initiate printing. After it is ended by a RETURN, a "lower right" coordinate must still be entered followed by a RETURN.

On the ATARI 825, the setup string

"ESC CTRL-T RETURN

sets the printer to 132 characters per line and the string

"ESC CTRL-S RETURN

returns the printer to 80 characters per line, its default setting. In both the CTRL-T and CTRL-S, the CTRL key is held down while the T or S is pressed.

The ability to print 132 characters per line can be very significant for wide VisiCalc sheets.

As an example of another use of a setup string, Figure 4-28 was printed at 132 characters per line, and every line was underlined. This was done by entering a setup string of

"ESC CTRL-T CTRL-O RETURN

To stop underlining, enter a setup string of

"CTRL-N RETURN

+, −, & Of these choices, the + is used to advance the paper one line before printing, and can be held down to advance multiple lines to separate one report from another. The − and & are used with serial devices as explained in the VisiCalc documentation.

Simple Budget	
Employees	50
Personnel	1000000
Benefits	250000
Telephone	2000
Rent	10000
Travel	15000
Hospitality	3000
Equipment	20000
TOTAL:	1300000

Figure 4-27. A single spaced printing of our report.

Simple Budget	
Employees	50
Personnel	1000000
Benefits	250000
Telephone	2000
Rent	10000
Travel	15000
Hospitality	3000
Equipment	20000
TOTAL:	1300000

Figure 4-28. An underlined printing of our report with condensed print.

The Print command prints all columns with the column width in effect, and it uses existing formats from the sheet. Labels and current values of the entry are printed, not the formulas. The screen may appear in an altered state while printing but it is restored by VisiCalc after printing.

On occasion we may want to stop printing before the full report has been printed. To do so, press down the CTRL key and, while holding it down, press the C key.

Considerations in Printing

There are a number of considerations in using the Print command to print a report to a printer.

- Each printer has a maximum number of characters per line that can be printed, for example, 132 on the ATARI 825. If we attempt to print rows that are wider than 132 on this printer, then the additional characters will print on the next lines. The capability for characters printed per line varies for different printers.
- If we want to print a report wider than the capability of our printer, we can do so by printing it in several sections and then combining them into one report. For example, if we have a report with twelve columns (A through L), each 15 characters wide and with 40 rows (1 through 40), we can print columns A through H (120 characters wide) by printing the rectangle between A1 and H40 and then printing again from H1 to L40. By printing column H in both reports, it may be easier to align them when overlaying one on the other for the final report.
- Although the window may be split when printing, this is ignored when the sheet is printed. However, different sections of the sheet may be selected and sequentially printed for the desired format.
- Even though we may have titles fixed in place, the rectangle from the cursor to the lower right coordinate is printed ignoring the appearance of these titles on the screen, that is, as if there were no fixed titles.
- In identifying the desired lower right location, a split screen can be helpful in locating the coordinate. Use the semicolon (;) to jump the cursor to the lower (or right-hand) screen and have it land on the desired lower right location (which we earlier left). Then by entering a RETURN, we will initiate the printing.
- Another method of identifying the lower right coordinate and simplifying the printing process is to include a row of directions like row 2 of the screen shown in Figure 4-29. Here we have a direction to go to A2 and then enter

/PP B20 RETURN

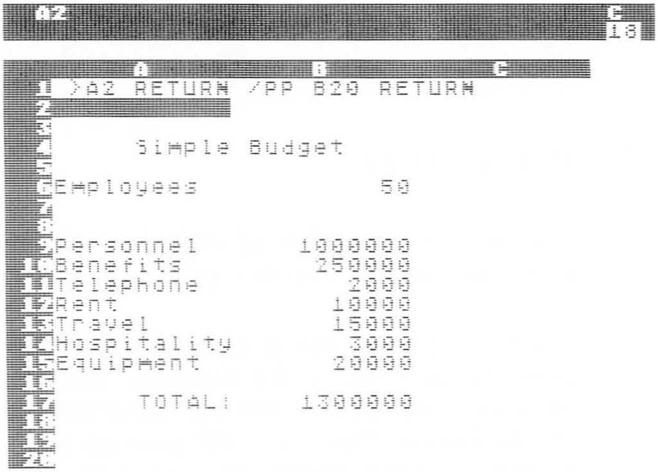


Figure 4-29. This spreadsheet contains directions included as a row of the sheet.

Following these directions will produce our report. This row has been entered as label data, beginning with the double quote ("). We cannot get VisiCalc to do this automatically for us, but we can include this as internal documentation, here serving as a reminder for us of the necessary steps to print our report. This can be especially helpful with setup strings.

• Notice in Figure 4-29 that we have left several blank rows at the top and at the bottom of the specified printing rectangle. On some friction feed printers, the top line of a report may be spaced too closely to the line beneath it. Placing blank rows at the top of the sheet eliminates this problem. The final blank rows eject the paper from the printer so that it is ready to be torn from the printer if a paper roll is used.

• If a report is to be prepared on special preprinted forms then an alignment check may be necessary. To do so, we can place a rectangle of ''dummy'' data on the top of the sheet and then print from this area above the desired report. This can be used to test that the paper has been properly loaded into the printer.

Chapter 8, Recognizing, Preventing and Correcting Errors, and Chapter 9, Creating Templates, also contain considerable information and suggestions regarding printing.

If desired, the following space can be used to record special procedures or setup strings for the printers used.

Printers	Setup strings

/R THE REPLICATE COMMAND

The Replicate command, /R, is used to replicate labels, values, or formulas from one section of the sheet to another. The examples in this section will explain the meaning of replicate as used in this context. The command is initiated by typing

/R

at which time a series of prompts asks us in turn for a SOURCE RANGE and then a TARGET RANGE. These are

SOURCE RANGE—the coordinates of the items which are already entered on the sheet and which we want to reproduce elsewhere.

TARGET RANGE—the coordinates of the locations into which we want our items replicated.

The Replicate command was introduced in some detail in Chapter 3, Getting Started; a rereading of that section may be helpful before continuing.

We must prepare a request in which the source and target ranges are each specified as a series, with the source first, then a colon, then the target.

Let's look in detail at some examples of replication processes, beginning with labels and studying what we want to do, why we want to do it, and then how we do it with VisiCalc. In each case refer to Figure 4-30.

Replicating Labels

Example 1.

What we want done: Copy the contents of one entry to another location, for example, from B2 to D2.

Why we might want to do this: We may have entered a lengthy label that we want to enter again completely at another location. Instead of typing it fully, we can use the Replicate command to do this for us.

How we do this: Let's assume we have the word YEAR at B2. We place the cursor at B2 and type

/R

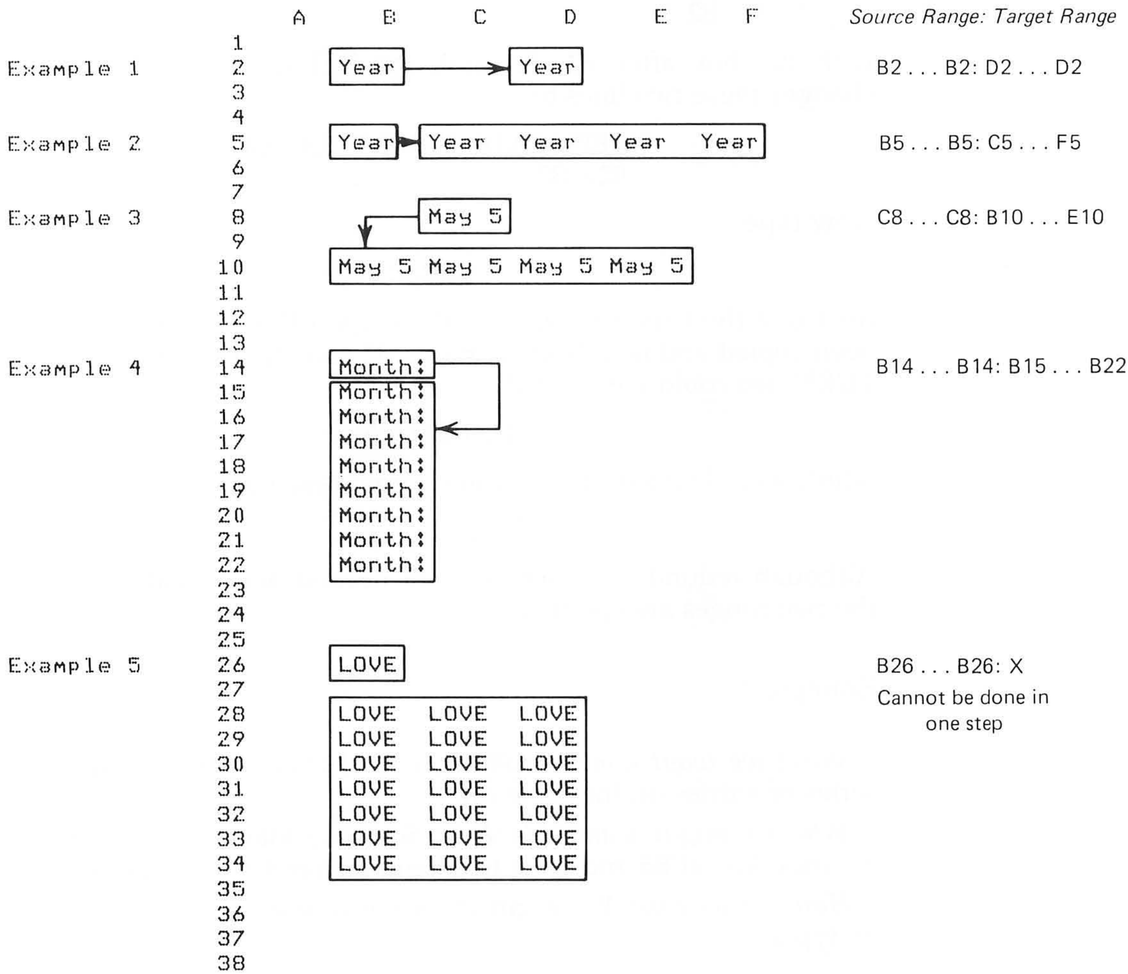


Figure 4-30. Examples of the process of replicating a single entry onto other areas of the sheet.

which generates prompt and edit lines as follows:

REPLICATE: SOURCE RANGE OR RETURN
B2

with the box after B2. Pressing the RETURN or colon (:) key changes these two lines to

REPLICATE: TARGET RANGE
B2...B2:

Now type

D2

(or move the cursor to D2) and then type RETURN. Our label has been copied and is in both locations B2 and D2. Instead of the RETURN, we could have typed

.D2 RETURN

which would give us a source and target range of

B2...B2:D2...D2

Although redundant (a series is not needed here), that's the way the two ranges are specified.

Example 2:

What we want done: Replicate a label from a single entry to a series of entries on the same row.

Why we might want this done: Suppose that we have entered an underline at B5 and want to extend it from C5 through F5.

How we do this: We begin the same way with the cursor at B5 by typing

/R:

then typing

C5.F5 RETURN

to complete the replication. We can use the arrow keys to enter the coordinates; after moving to C5, the period (.) jumps the cursor back to B5 as the RETURN does after entering F5.

Example 3:

What we want done: Replicate a single entry across a row, as shown in Example 3 of Figure 4-30.

Why we might want to do this: We may have entered a date at the top of the sheet that we want to repeat at the top of each column.

How we do this: With the cursor at C8 type

/R:B10.E10 RETURN

If the cursor is not at C8 to begin with, we can then type, without moving it,

/R BACK S C8:B10.E10 RETURN

The BACK S erases the current cursor location.

Here, as in other uses of the cursor to identify a series, if we have used BACK S to erase the current location and then decide that we do want it, we can obtain it again on the line by typing the space bar.

Example 4:

What we want done: Replicate a single label entry down a column.

Why we might want to do this: We want to repeat a label, as shown in Example 4.

How we do this: As in the previous examples, we place the cursor at B14 and type

/R:B15.B22 RETURN

Example 5:

What we want to do: Replicate a single label entry into a two-dimensional shape.

How we do this: This cannot be done with one Replicate command. To accomplish this, we'll need several replicate commands.

In all these examples, we began with a single entry. Now let's begin with a row and replicate a row, also of labels. Refer to Figure 4-31 for these examples.

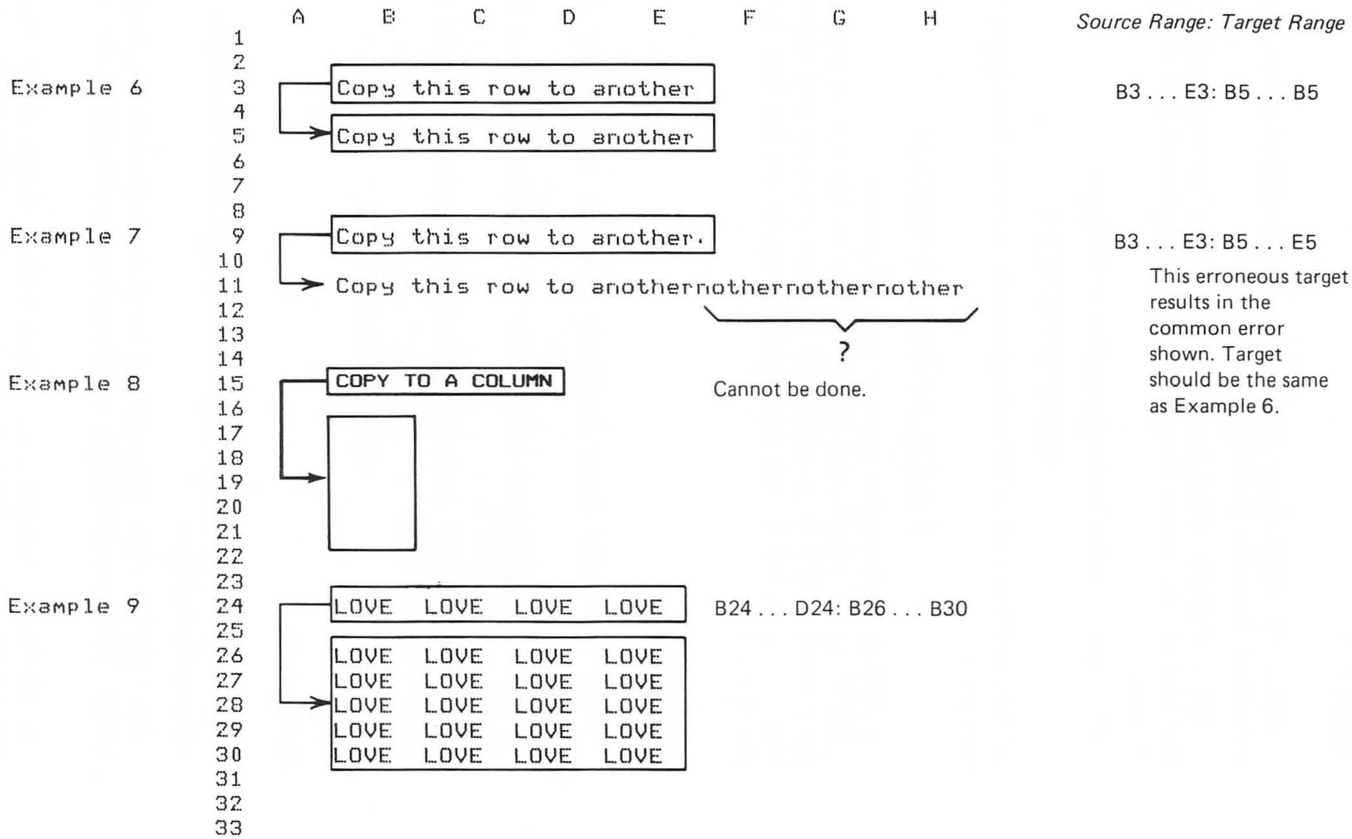


Figure 4-31. Examples of the process of replicating a row onto other areas of the sheet. The same principles apply to replicating a column.

Example 6:

What we want to do: Copy a row into another row.

How we do this: The command to do this, with the cursor beginning at B3, is

/R.E3:B5.B5 RETURN

This is tricky, and the cause of many errors. When replicating a row (or column) into another row (or column), only the starting coordinate of the target is provided. VisiCalc does the replicating with only this information. In the next example we'll see what happens if we give both the start and end of the target range.

Example 7:

What we want to do: Same as example 6.

How we do this: Now let's put the cursor at B9 and enter

/R.E9:B11.E11 RETURN

Our results are in error. This kind of specification is used only to replicate a row (or column) into a two-dimensional area as we'll see next.

Example 8:

What we want to do: Replicate a row down a column.

How we do this: This cannot be done with this command. We'll see next that a horizontal source range with a vertical target range results in a two-dimensional area, not in perpendicular areas.

Example 9:

What we want to do: Replicate a row into a two-dimensional area.

How we do this: With the cursor at B24, we enter

/R.E24:B26.B30 RETURN

In this example we get the row successfully repeated over several rows.

These concepts for replicating a row apply similarly to replicating a column.

Finally let's discuss replicating a two-dimensional area of labels. This cannot be done in one step with the Replicate command although repeated use of this command can do this task.

Replicating Formulas With Relative Coordinates

When formulas including coordinate references are replicated, one step is added to the process of replicating. Let's follow this processing, beginning with the screen shown in Figure 4-32. Here we have entered numbers for our sales for each quarter of last year and this year. We've placed a total at the bottom of each of these two columns and we're beginning the process of finding the difference between sales this year and last year.

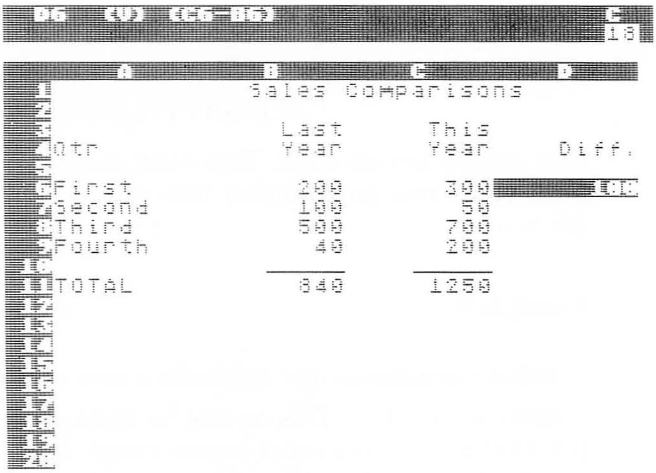


Figure 4-32. We'll replicate the formula at D6 down column D.

Notice that at D6 we have the formula

(C6-B6)

We want similar, but not identical, formulas all the way down that column. What we want is

Location	Formula
D7	(C7-B7)
D8	(C8-B8)

D9	(C9-B9)
D11	(C11-B11)

The similarity between these formulas occurs because each performs a computation like the one at D6 but with coordinates relative to location.

The power of the Replicate command allows the replication process to occur while accommodating these changing relationships. Let's see how.

We'll begin the process with the cursor at D6 by typing

/R:D7.D11

which establishes a source and target range of

D6...D6:D7...D11

Now we'll press RETURN, producing the edit line

D6: D7...D11: (C6

with the C6 appearing in a box. Now the prompt line asks us to enter either an N or an R. It says

REPLICATE: N=NO CHANGE, R=RELATIVE

In our discussion we've emphasized that these coordinates are relative. We'll enter an

R

here. The prompt line is unchanged, in effect again asking us to enter an N or an R. However, the edit line now shows

D6: D7...D11: (C6-B6

with B6 in a box. Again we'll enter

R

meaning that B6 is considered a relative coordinate down this column.

As we indicate that B6 is relative (by pressing R), VisiCalc has all the necessary information to replicate the item down the column; we now have the screen of Figure 4-33.

Everything is correct, with the exception of D10, where 0 (zero) has occurred as the difference between the two labels at C10 and B10. This illustrates the desirability of postponing parts of the labeling process until the last steps of preparing electronic sheets. We'll discuss this again at several points in later chapters.

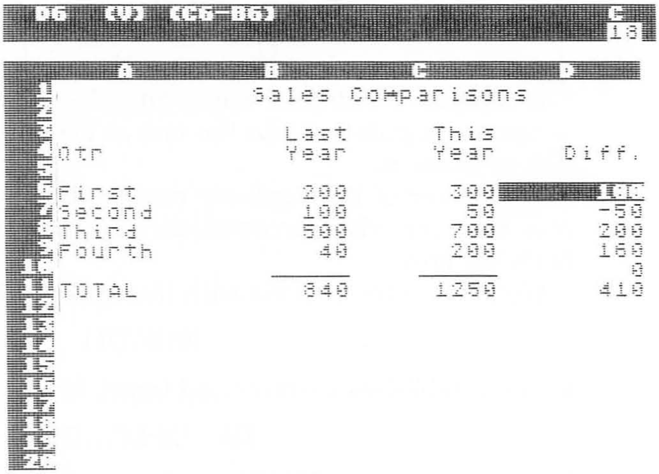


Figure 4-33. We have successfully replicated our formula with the exception of entry D10, where we'll need to correct the entry.

Replicating Formulas With No Change Coordinates

Let's continue with a similar example and explain how the NO CHANGE is used. We've simplified our report and have one column for sales THIS YEAR and another column started in column D in which we want to determine the percentage of our sales in each quarter of the year, as shown in Figure 4-34. At location D5 we've placed the formula

(B5/B9)*100

that is, divide the sales for the first quarter by the total and multiply by 100. Notice that an integer format (/FI) has been placed at the entry.

Now consider the formulas that we want at the other locations. What we want is

Location	Formula
D6	(B6/B9)*100
D7	(B7/B9)*100
D8	(B8/B9)*100

In studying the formulas, notice that the first coordinate changes relatively as we go down the column; however, B9 is absolute, it does not change; that is, it is NO CHANGE.

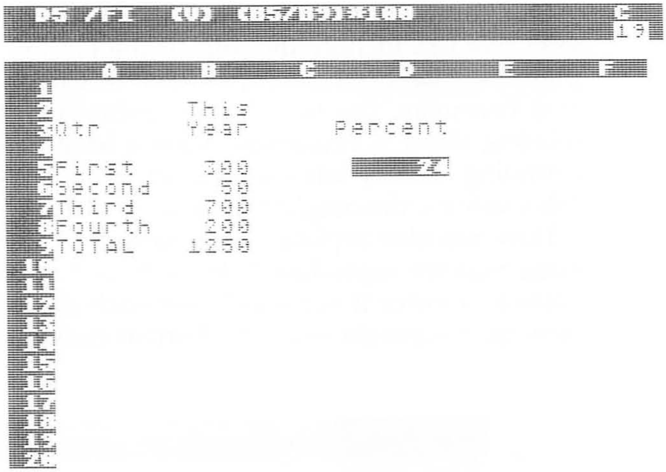


Figure 4-34. We'll replicate this formula, using both the RELATIVE and NO CHANGE options.

Let's replicate. The cursor is at D5.

We type	Meaning
/R:	Initiate the Replicate command and establish this as a replication of one entry.
D6	Type this as shown or move the cursor to this location.
.	Signals the middle of the series and adds an ellipsis (...) to the line.
D8	Enter the end of the series.
RETURN	End the TARGET RANGE. This places the formula on the edit line with B5 highlighted and prompts us for an R for relative or an N for no change.
R	This is the relative part of the formula.

We're then similarly prompted for B9, which appears under the highlight.

N	This entry, B9, does not change. It's the same in each formula.

Our formulas are replicated as required on the screen of Figure 4-35. We can include the total at the bottom of column D and complete the sheet. (*A word of caution:* In Chapter 5, Labels, Numbers, and Formulas, the topic of computing percentages and accurately totaling them is discussed. Values have been chosen here so that rounding occurs with a total of 100%. With different quarterly sales values, this might not occur.)

Here we also replicated the format by including it with the formula that we reproduced. If we hadn't done that we would be required to enter it separately for each entry since VisiCalc doesn't provide a separate Replicate-Format command.

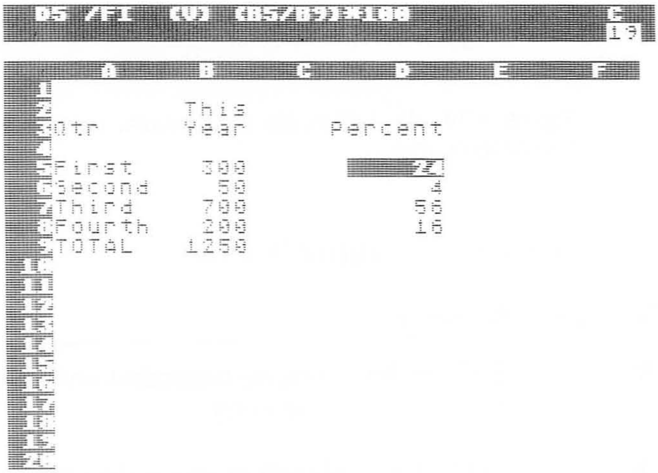


Figure 4-35. We have replicated D5 (including the format) down column D as planned.

Additional Considerations

This command is the most complex of the VisiCalc commands and can easily lead to significant problems in developing a sheet. For this reason, when using this command experimentally, it's important to copy the current sheet to backup so that if replicating causes problems, the previous sheet can be retrieved. Backup is discussed in the next section of this chapter and in Chapter 7, Other Topics.

Finally, this command can be used to erase portions of a sheet by replicating blank rows onto two-dimensional areas.

/S THE STORAGE COMMAND

The Storage command, /S, provides the capability to save, load, and delete files on diskettes or other media in several formats, to initialize a diskette, and to exit from the VisiCalc program. It is initiated by typing

/S

at which time the prompt line reads

STORAGE: L S D I Q #

These correspond with the following options:

L	Load a file into memory.
S	Save a file from memory onto storage media.
D	Delete a file.
I	Initialize a diskette.
Q	Quit (exit) the VisiCalc program.
#	Save or load a file in the Data Interchange Format, DIF (trademark of Software Arts, Inc.).

We'll briefly discuss each after presenting some general considerations of this command.

General Considerations

In working with this command, a number of factors can be applied to all of the Storage options.

- Files are stored for a variety of reasons. We may want to reuse a spreadsheet at a later time with different values. We may be trying a new command, or a new formula, and want to preserve the sheet momentarily while we experiment. We may be creating a backup or duplicate copy, which can be used if we inadvertently destroy the original. We may wish to stop work on a sheet that we've only partially finished and will finish later. We may have accidentally pressed the SYSTEM RESET key at which time VisiCalc requires us to save the current status of the sheet. For these reasons, and others, this is an important command.
- VisiCalc works with a single spreadsheet at a time. That sheet must be in memory. If it's a sheet on a diskette, we must load it into memory before VisiCalc can be used with it.

• If we change a sheet in memory, we do not change the corresponding sheet if it is stored on a diskette. Thus, we can manipulate a sheet in memory as desired without affecting the stored sheet.

• VisiCalc accesses disk storage only under our control. If we revise a sheet, it is not changed on the diskette until we act to store it again.

• Files have unique names that must be eight characters or less in length, with a letter as the first character. Other characters can be letters or numbers only (no other characters are allowed).

• Prefixes may be used with filenames to specify a device type. Four are permitted:

Dn: To identify a specific disk drive; for example, D2:FIG42 identifies file FIG42 as being on a diskette currently in drive 2. If drive 1 is used alone this prefix is unnecessary.

C: To identify a file on cassette, as specified in the VisiCalc documentation.

Rn: To be used with RS-232 devices as specified in the VisiCalc documentation.

P: To print the formats and formulas of the sheet as described in detail below.

• Suffixes are used after the filename. VisiCalc appends .VC to files stored as full spreadsheets. This is done automatically and we don't have to type these characters when loading or saving the file. Other suffixes of .PRF and .DIF are used for /PF files and files stored with the # option respectively.

• The * can be used in conjunction with a suffix in accessing filenames. For example, suppose that we want to scroll filenames of a diskette with many files including some that start with FIG, for example FIG42, FIG43, etc. By typing FIG*.VC as the file name, and then scrolling, we'll have only those filenames beginning with FIG appear on the edit line.

• When prompted by VisiCalc to enter a file name (with a prompt such as FILE TO LOAD, FILE FOR SAVING, or FILE TO DELETE), we can enter the file name in one of two ways:

1. Respond by typing the file name, followed by a RETURN.
2. Respond by pressing the right arrow key. This causes VisiCalc to access the diskette and print on the edit line the name of the first file stored. Repeatedly pressing this key causes the names of the files on this diskette to scroll one by one on the prompt line. When the desired name ap-

pears, pressing the RETURN key (not the right arrow key) will perform this action on the file. Continued scrolling results in leaving the command.

This is a very convenient feature since it can save typing keystrokes and it helps us if we cannot remember the name of the file that we want.

• The RETURN key and the right arrow keys are located close together and therefore caution must be used when scrolling so that the desired action occurs appropriately.

• An error can occur for a number of reasons; for example, no diskette is in the disk drive, no file is on the diskette with a name that we supply, etc. When this occurs, we receive the general message

ERROR: NOT FOUND
or ERROR: I/O

from VisiCalc without additional details of the cause of the error.

• Once a file name appears on the edit line—for example, as a result of scrolling the diskette directory, or list of file names—we can use the BACK S key to edit the name. For example, if we want to load FIG22 from the diskette and the prompt line now shows FIG12.VC, we can press BACK S five time, type 22, then RETURN, and get the desired file loaded. We don't need to type the .VC since VisiCalc will do this automatically for us.

• Naming files with some consistent format can save time. For example, the files used for examples in this book are named FIG41, FIG42, etc. If we want to load a file, we can get the first name from the directory and edit the last few characters. Thus, we are able to use meaningful names without the necessity of typing them fully each time, or the necessity of scrolling the full directory.

• The first file on the diskette can be a file created by simply storing an empty sheet with a desired name. There are two reasons for doing so. First, as stated, we can establish a file with the name

FIG4

When we scroll, we simply type the rest of the name. Second, if we are doing a large amount of file accessing in one sitting, it's possible to act hurriedly and inadvertently delete a file or save one file over another without scrolling to the correct file name. This often occurs with the first file listed. Therefore, if we put a dummy, or unused, file first, it can on occasion be the deleted file or the file over which we've saved another without causing any harm.

- The BREAK is used to cancel these actions.

The Storage Command Options

Now we can examine each of the options of the Storage command.

L—Used to load a file from a diskette into memory. We are prompted for the file name, which we type or enter by scrolling the directory. The screen should be cleared first unless we want to overlay one sheet on another.

The overlay facility places one sheet over another, with the labels, values, and formulas of the second destroying overlaid entries of the first. Blanks on the second do not change the first.

This capability can be important where, for example, we have data that we may wish to use in more than one sheet. If we receive monthly data on an activity and we use it in several reports, we can enter that data in column A (any location will be satisfactory) and then build several reports that use these data in this format in column A. We load a report format, then the data. Recalculate (!) and we have the report. Clear the sheet, load the second report, then the data again. Recalculate and we have the second report, etc.

When a file is loaded, it appears with the same characteristics (column width, cursor location, etc.) with which it was stored.

S—This option saves the sheet with all labels, numbers, and formulas intact. The current coordinate entries and cursor location on the screen are saved as well as the global characteristics, so that when loaded we have the same screen as when it is stored.

We provide the file name by typing it or scrolling to it. If we save a file with a name that is already in the directory, we destroy the file on the diskette by writing the sheet currently in memory over it.

D—This option deletes a file from the diskette. Since this is a final action (the file cannot be retrieved with VisiCalc), we are prompted to confirm by entering a Y, with most other keys canceling. Again we provide the name of the file that we want deleted by typing it or scrolling to it.

I—The initialize option prepares a diskette to store files. In doing so, any previously written files become inaccessible. If we have a system with several disk drives, we should verify that we are using the desired drive. This is indicated by the D number on the edit line. Change the number if necessary (using BACK S and then typing the correct value). Again we are asked to confirm this action, here by typing the RETURN key.

Q—This allows us to quit, or exit, from VisiCalc and start up with another non-VisiCalc program. We respond by entering a Y to confirm that we want to quit, or the BREAK key to cancel. The effect of quitting is the same as if we turn the power off and then on with a new diskette in place.

#—This option saves or loads a file using the Data Interchange Format, DIF (trademark of Software Arts, Inc.). It includes two options,

S Save a file
L Load a file,

and within these R (row) and C (column) options.

The "Programmer's Guide to the Data Interchange Format," referenced in the Bibliography, contains detailed information about the format used to store this type of file.

When a file is saved, with this option any formulas on the sheet are not saved. Instead, the values of these entries are saved, as well as any other numbers and labels. For values, the full number of significant digits computed by VisiCalc is stored.

When saving a file with this option, a rectangular area is saved. We place the cursor at the top left and specify the lower right, as we did with the Print command. The resultant rectangle is saved, by row or column as we specify.

When loading, we place the cursor at the top left corner of the area on the sheet where we want the rectangle loaded. This provides another way to combine sheets, since more than one file can be loaded into memory.

Additional "Storage" Option: Print the Contents of the Sheet in an As-Entered Format

We can use the storage command to get a full listing of our formulas, labels, values, global formats, etc., of the sheet on which we are working. To do so we type

/SSP: RETURN

Let's look at an example. Suppose that we are working with the sheet of Figure 4-36 and that we want to get the listing that we have described. Figure 4-37 shows the listing after typing that line. If we want to, we can reproduce the full sheet by typing the lines shown in Figure 4-37.

Notice in Figure 4-37 that the entries are listed in "reverse" order from the bottom right, here C11, to the top left, here A1. Following A1, we see a series of commands that sets up the sheet when this file is loaded.

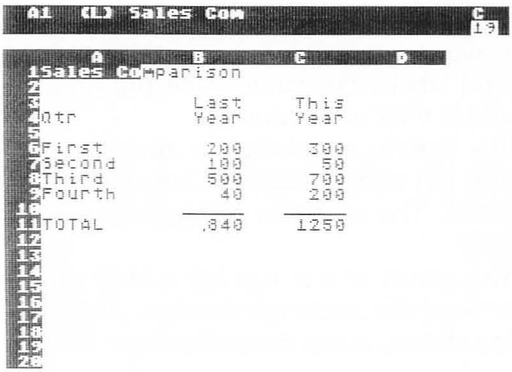


Figure 4-36. We've entered this report and now want to print all of the underlying relationships, as shown in Figure 4-37.

```
>C11:@SUM(C6...C9)
>B11:@SUM(B6...B9)
>A11:"TOTAL
>C10:/FR"______
>B10:/FR"______
>C9:200
>B9:40
>A9:"Fourth
>C8:700
>B8:500
>A8:"Third
>C7:50
>B7:100
>A7:"Second
>C6:300
>B6:200
>A6:"First
>C4:/FR"Year
>B4:/FR"Year
>A4:"Qtr
>C3:/FR"This
>B3:/FR"Last
>B1:"mparison
>A1:"Sales Co
/W1
/GOC
/GRA
/GC8
/X>A1:>A1:
```

Figure 4-37. The printed report resulting from typing /SSP: RETURN with the sheet of Figure 4-36 in memory.

Command	Means
/W1	Start with one window.
/GOC	The order of recalculation is columnwise.
/GRA	The recalculation mode is automatic.
/GC8	The column width is 8.
/X>A1:A1	This unusual series of characters is a command used to set up the portion of the sheet displayed on the screen. The /X initiates a command, here >A1:>A1: This moves the cursor first to A1, then redundantly to A1. This is explained in the next example.

In another example we have the problem of Figure 4-38, which is the same sheet as earlier but displayed with a split screen. Our use of

/SSP: RETURN

produces the same listing as the previous example with the exception of the last few lines, shown in Figure 4-39.

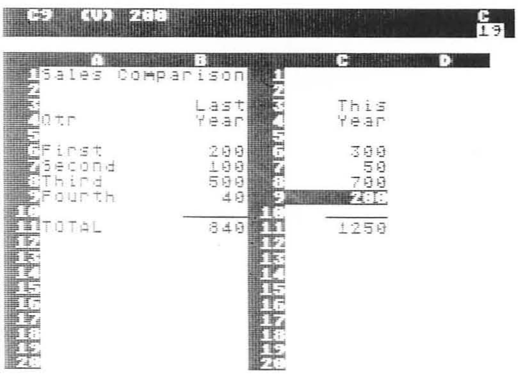


Figure 4-38. A split screen sheet for which we'll get a printed list of numbers, labels, and formulas.

```
/W1
/GOC
/GRA
/XV19
/GC8
/X>A1:>B1::/GC8
/X>C1:>C9:
```

Figure 4-39. A partial list of the report resulting from typing /SSP: RETURN with the sheet of Figure 4-38 in memory.

Of interest are the following lines with their interpretation

Command	Means
/XV19	Split the screen vertically at the 19th character from the left.
/GC8	Establish a global column width of 8 in the left window.
/X>A1:>B1:;/GC8	The first portion of this, until the second colon, is similar to the example above. Then the semicolon (;) jumps the cursor to the other screen, where a column width of 8 (/GC8) is also established.
/X>C1:>C9	This command now controls the portion of the screen shown on the right side of the split screen. By jumping to C1 it places the portion of the sheet we want on the screen and then moves the cursor to C9.

All of this usage occurs automatically as the file is loaded; however, these /X commands will work if we enter them normally from the keyboard. All can be accomplished in easier, more direct ways, but this can be potentially helpful in creating input files, which are then loaded into memory by VisiCalc.

/T THE TITLES COMMAND

The Titles command, /T, is used to fix column or row titles in place above or at the left of the screen. The command is initiated by typing

/T

at which time the prompt line reads

TITLES: H V B N

prompting us as follows.

Option	Means
H	Horizontal title only.
V	Vertical title only.
B	Both horizontal and vertical titles.
N	No titles.

The value of this command can be seen in Figure 4-40, where the large budget in the background is one electronic sheet. Two screens have been imposed upon this background. The top screen simply shows a section of the screen without any identification labeling the rows or columns displayed. In the bottom screen, however, we have used the Titles command to display the row labels (columns A and B) on the left of the screen. This command allows us to freeze columns, as in the bottom screen of Figure 4-40, where titles are frozen vertically.

In Figure 4-41 the same budget is displayed with titles frozen both vertically and horizontally (B).

Some general considerations in using this command are

- The location of the cursor when the command is executed is important. With the H option the row of the cursor location and all above it are frozen. With the V option the column of the cursor and all to the left of it are frozen. With the B option the rows and columns above, to the left, and including the cursor location are frozen.
- Frozen titles are released by entering the N option.
- Once frozen, the rows or columns become the new edge of the sheet and the cursor will beep if we try to move into this area with an arrow key.

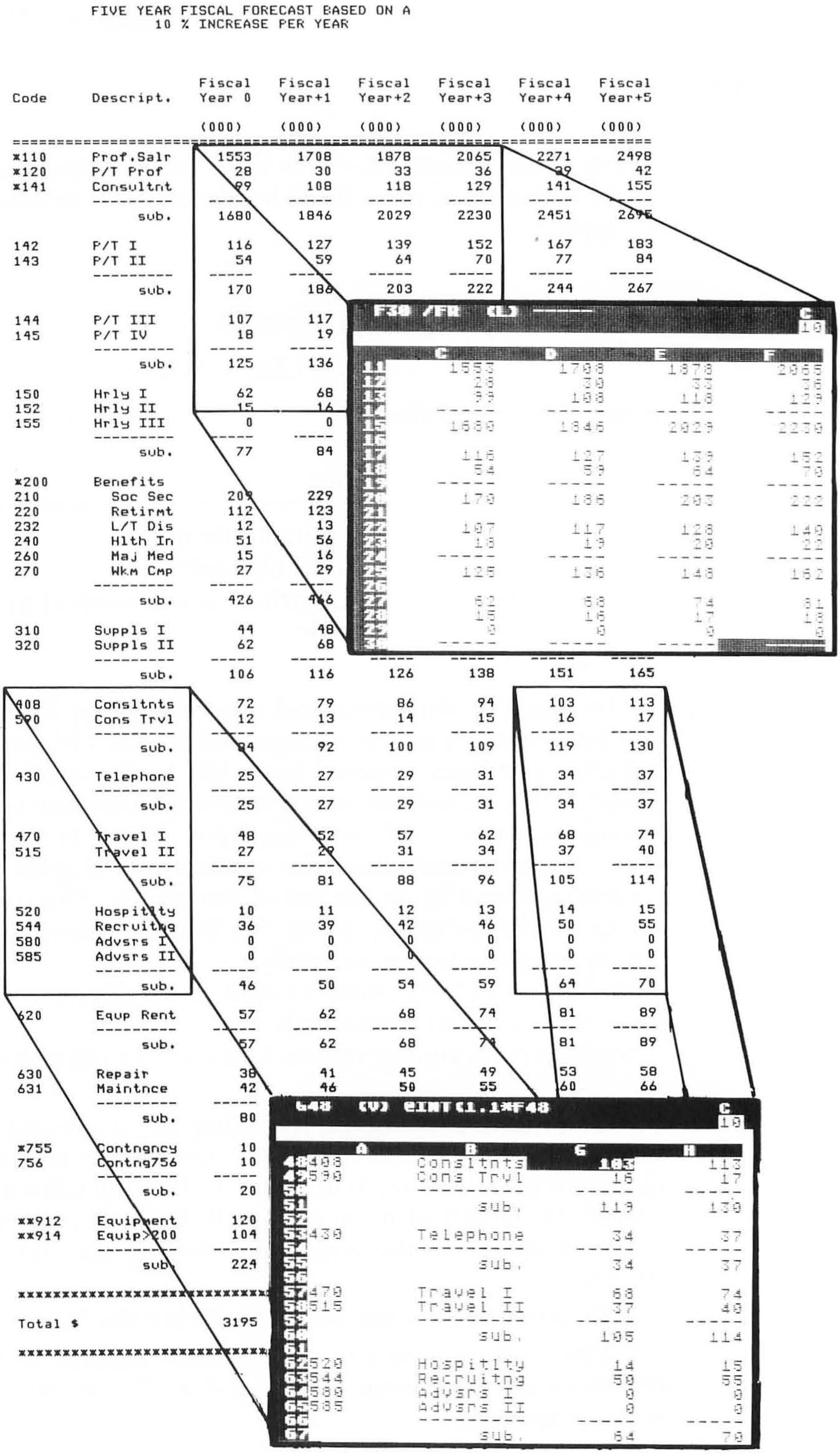


Figure 4-40. A demonstration of a screen without titles (at the top) and another in which frozen titles improve the readability of the screen.

FIVE YEAR FISCAL FORECAST BASED ON A
10 % INCREASE PER YEAR

Code	Descript.	Fiscal Year 0 (000)	Fiscal Year+1 (000)	Fiscal Year+2 (000)	Fiscal Year+3 (000)	Fiscal Year+4 (000)	Fiscal Year+5 (000)
*110	Prof.Salr	1553	1708	1878	2065	2271	2498
*120	P/T Prof	28	30	33	36	39	42
*141	Consultnt	99	108	118	129	141	155
	sub.	1680	1846	2029	2230	2451	2695
142	P/						183
143	P/						84
							267
144	P/						169
145	P/						26
							195
150	Hr						97
152	Hr						20
155	Hr						0
							117
*200	Be						
210							333
220							178
232							17
240							80
260							20
270							40
	sub.	426	466	509	558	610	668
310	Suppls I	44	48	52	57	62	68
320	Suppls II	62	68	74	81	89	97
	sub.	106	116	126	138	151	165
408	Consltnts	72	79	86	94	103	113
590	Cons Trvl	12	13	14	15	16	17
	sub.	84	92	100	109	119	130
430	Telephone	25	27	29	31	34	37
	sub.	25	27	29	31	34	37
470	Travel I	48	52	57	62	68	74
515	Travel II	27	29	31	34	37	40
	sub.	75	81	88	96	105	114
520	Hospitlty	10	11	12	13	14	15
544	Recruitng	36	39	42	46	50	55
580	Advsrs I	0	0	0	0	0	0
585	Advsrs II	0	0	0	0	0	0
	sub.	46	50	54	59	64	70
620	Equp Rent	57	62	68	74	81	89
	sub.	57	62	68	74	81	89
630	Repair	38	41	45	49	53	58
631	Maintnce	42	46	50	55	60	66
	sub.	80	87	95	104	113	124
*755	Contngncy	10	11	12	13	14	15
756	Contng756	10	11	12	13	14	15
	sub.	20	22	24	26	28	30
**912	Equipment	120	132	145	159	174	191
**914	Equip>200	104	114	125	137	150	165
	sub.	224	246	270	296	324	356
Total $		3195	3501	3834	4204	4610	5057

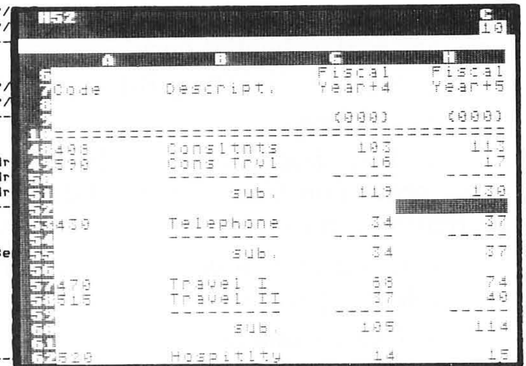


Figure 4-41. A screen in which titles have been frozen in both (B) directions.

• To move into a frozen area, the GO TO command must be used.

• The status of any frozen titles is maintained with the sheet if the sheet is stored.

• Titles can be frozen separately in split screens. This can be an important capability since it allows us to display four separate areas of the sheet at once.

• Areas frozen into titles can include values and formulas as well as labels.

• Areas frozen with the Titles command have a modest amount of protection from accidental destruction and can be important for that reason as explained in Chapter 9, Creating Templates.

• Using titles reduces the area of the screen that can be used to scroll the spreadsheet.

• This command only affects the display and not the contents of the sheet.

• If the sheet is printed, titles that have been frozen are ignored, and the rectangle from cursor to lower right is printed.

• Our actions may necessitate the removal of frozen titles by VisiCalc. For example, if we issue a GO TO for an entry behind frozen titles the GO TO has precedence and the titles will be removed and not reinstated.

/V THE VERSION NUMBER COMMAND

The Version Number command, /V, produces a prompt line that contains the version number for this copy of VisiCalc. The examples in this book were prepared on an ATARI 800 with version number 1.74A.

Computer software often is revised as problems or errors are detected or enhancements are made.

/W THE WINDOW COMMAND

The Window command, /W, is used to split the screen either horizontally or vertically into two windows that can be scrolled separately. It is initiated by typing

/W

at which time the prompt line reads

WINDOW: H V 1 S U

indicating the available options. They are

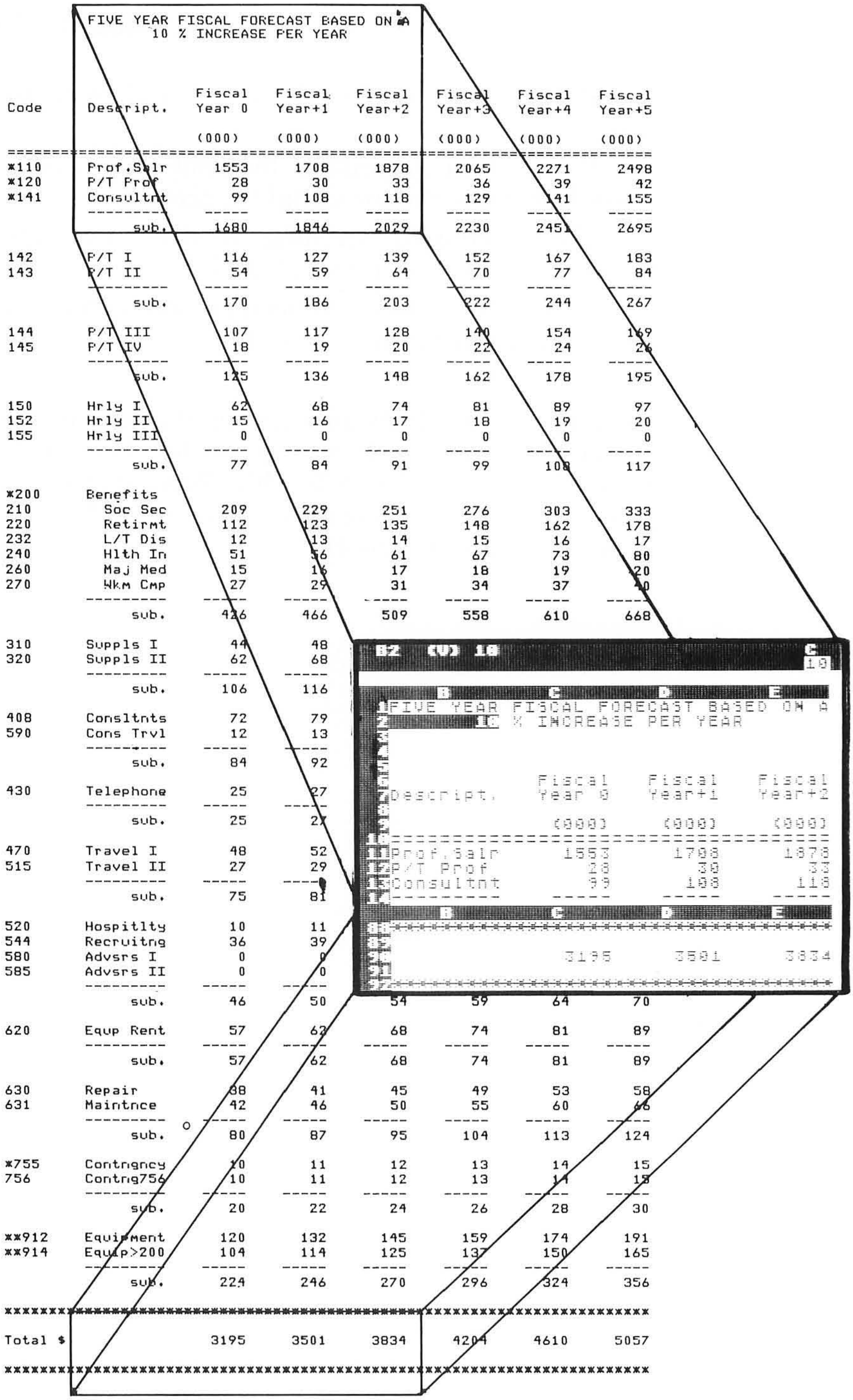
Option	Means
H	Split the screen horizontally.
V	Split the screen vertically.
1	Return from a split screen to a single screen.
S	Scroll both windows simultaneously (synchronized).
U	Discontinue synchronized scrolling, that is, unsynchronize the scrolling action.

Some general considerations when using this command are

- The command affects the display and not the contents of the sheet.
- There continues to be only one sheet, not two separate sheets, so that changing an entry in one window affects both windows.
- With the H option, a single row of column labels is included in the display above the current cursor location, and the cursor is moved up one row into the top half of the screen. One row is removed from the bottom screen, and the contents of the bottom screen are moved down one row to accommodate the new row.
- With the V option, a new set of row labels is included in the three character positions to the left of the current cursor location. The cursor is moved into the left screen on the same row.
- With a split screen, it's possible to have separate global formats and column widths in each window.
- It's possible to freeze titles separately in each window, providing an opportunity to see four separate areas of the sheet at once.
- The semicolon (;) causes the cursor to jump from screen to screen, landing at its most recent position on the screen to which it moves.

- If a sheet is stored while split, it will be reloaded in the same state.
- If the sheet is printed, the split screen is ignored, and the sheet from cursor to lower right is printed.
- Unsynchronized scrolling is used by default unless the Synchronized (S) option is selected.
- Once the S option has been chosen, enter U to turn it off and return to unsynchronized scrolling.

Figure 4-42 contains an example of a screen superimposed on a larger sheet in which the screen has been split and windows scrolled separately in each. This allows us to visualize simultaneously the effect of changes in the items at the top upon the totals at the bottom of the sheet.



FIVE YEAR FISCAL FORECAST BASED ON A
10 % INCREASE PER YEAR

Code	Descript.	Fiscal Year 0 (000)	Fiscal Year+1 (000)	Fiscal Year+2 (000)	Fiscal Year+3 (000)	Fiscal Year+4 (000)	Fiscal Year+5 (000)
*110	Prof.Salr	1553	1708	1878	2065	2271	2498
*120	P/T Prof	28	30	33	36	39	42
*141	Consultnt	99	108	118	129	141	155
	sub.	1680	1846	2029	2230	2451	2695
142	P/T I	116	127	139	152	167	183
143	P/T II	54	59	64	70	77	84
	sub.	170	186	203	222	244	267
144	P/T III	107	117	128	140	154	169
145	P/T IV	18	19	20	22	24	26
	sub.	125	136	148	162	178	195
150	Hrly I	62	68	74	81	89	97
152	Hrly II	15	16	17	18	19	20
155	Hrly III	0	0	0	0	0	0
	sub.	77	84	91	99	108	117
*200	Benefits						
210	Soc Sec	209	229	251	276	303	333
220	Retirmt	112	123	135	148	162	178
232	L/T Dis	12	13	14	15	16	17
240	Hlth In	51	56	61	67	73	80
260	Maj Med	15	16	17	18	19	20
270	Wkm Cmp	27	29	31	34	37	40
	sub.	426	466	509	558	610	668
310	Suppls I	44	48				
320	Suppls II	62	68				
	sub.	106	116				
408	Consltnts	72	79				
590	Cons Trvl	12	13				
	sub.	84	92				
430	Telephone	25	27				
	sub.	25	27				
470	Travel I	48	52				
515	Travel II	27	29				
	sub.	75	81				
520	Hospitlty	10	11				
544	Recruitng	36	39				
580	Advsrs I	0	0				
585	Advsrs II	0	0				
	sub.	46	50	54	59	64	70
620	Equp Rent	57	62	68	74	81	89
	sub.	57	62	68	74	81	89
630	Repair	38	41	45	49	53	58
631	Maintnce	42	46	50	55	60	66
	sub.	80	87	95	104	113	124
*755	Contngncy	10	11	12	13	14	15
756	Contng756	10	11	12	13	14	15
	sub.	20	22	24	26	28	30
**912	Equipment	120	132	145	159	174	191
**914	Equip>200	104	114	125	137	150	165
	sub.	224	246	270	296	324	356
Total $		3195	3501	3834	4204	4610	5057

Figure 4-42. A split screen in which the two windows allow us to display the effect on the totals of changing values at the top of the sheet.

/- THE REPEATING LABEL COMMAND

The Repeating Label command, /-, causes entered characters to be repeated to fill the field. It is initiated by typing

/-

at which time the prompt line reads

LABEL: REPEATING

We then type what we wish to have repeated across this single field. For example if, after typing /-

We type	The field is filled with
-	---------------
=	======
=-	=-=-=-=-
o	ooooooooo
A B	A B A B A

If the global column width changes, the displayed characters are readjusted to fill the field.

In each case the command itself followed by the characters entered is stored as the contents of the entry. As an example, if we type the third line above, the entry contents line contains

(/) =-

Fields entered with this command can be copied into other areas with the Replicate command.

/X UNNAMED

The /X is used by VisiCalc when reloading a sheet to split the screen (if appropriate), to position the sheet on the window, to freeze titles (if appropriate), and to position the cursor. This was discussed briefly in the Storage command section of this chapter. After entering /X, we can enter

H or V followed by a number and a RETURN to indicate where to split the screen.

As suggested in the discussion of the Storage command, this may also be useful for initiating other commands automatically in files prepared as input to VisiCalc.

All functions performed with the /X are more easily performed in other ways.

Chapter 5

Labels, Numbers, and Formulas

INTRODUCTION

This chapter provides information on entering and using labels, numbers, and formulas. In each case the necessary capabilities, limitations, and cautions are presented. We'll start by discussing labels.

LABELS

The variety of VisiCalc data appearing in the window of Figure 5-1 illustrates a number of the considerations involved in using labels. This entire window is composed of two lines of report titles at the top and then the row and column labels. Vertical lines separating the columns have been drawn on the figure for illustration.

Let's start at entry A5, where the letters QTR appear at the left of the column. To enter them, we simply type the three letters followed by a RETURN (or by an arrow key). When the Q is entered, the word LABEL appears on the prompt line. Entering any letter (A-Z) as the first character of an entry causes this. When we're done (we've pressed the RETURN key), the entry line shows

A5 (L) QTR

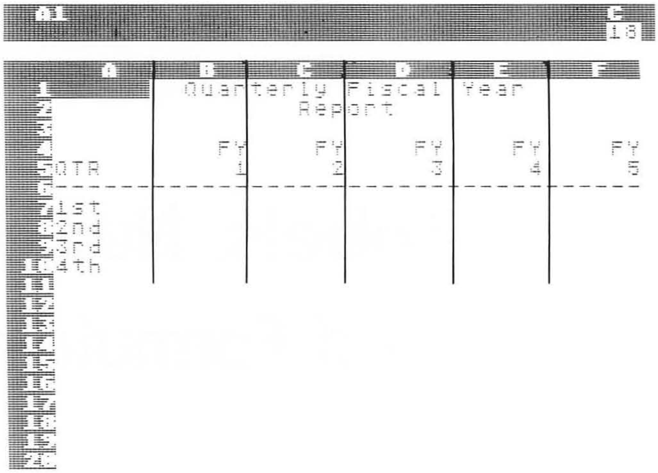


Figure 5-1. An illustration of a variety of labels on a VisiCalc window.

with the (L) indicating that a label has been entered. Notice that the label is at the left of the column. This is called left justification.

The top row of the screen illustrates several additional topics in dealing with labels. At location B1 we have the letters QUAR and at C1 we have TERLY. This division of labels is a common occurrence with VisiCalc.

If we typed the full word at one entry, it would be accepted and stored, but only a part of it would be displayed, depending on the column width. This limitation on column width means that we must often break labels in this way.

If we are working on a large sheet on which there are many labels and on which we may change column widths several times before we're satisfied, it's a good idea to postpone labeling until the last step. This can help to reduce the generally awkward work necessary to enter or reenter labels of this type on the sheet.

Also notice that at B1 the label QUAR is at the right of the column, right justified, not left justified as QTR was at A5. This occurs because the /FR is used at entry B1. This alternate formatting is also common in labeling.

When labeling, entering the format before the label is a good habit because it means that the label can then be ended with either a RETURN or an arrow key. If the format is entered after the data, it must be preceded by a RETURN (to end the label) and followed by an arrow, adding one keystroke.

On the same row, at location E1 we have the word YEAR in the middle of a column with a space before and after it. VisiCalc does not have a center format. For that reason we must enter it left justified with a blank before it or right justified with a blank after the word. However, if we try to enter a space before it by pressing the space bar nothing occurs. To enable us to enter the space to the left of the word YEAR, we must first press the double quote ("). This allows us to begin a label with a character other than a letter. Here we will press the double quote, then the space bar, then the word YEAR. That is, we enter

" YEAR

Notice that we do not use another double quote to close the line. If we do so, the quote appears on the line. Although this is desired for some labels, we do not want it here.

Another way to enter this same label is to enter the word YEAR followed by a space. After we begin a label, the space bar does place a space in the label. With this form of entry, we will need to use a right justification, / FR, to move the label, with the one space after it, to the right of the column.

The word REPORT on row 2 was also entered in parts, REP was right justified in C2 and ORT placed in D2. The label FY in B4 was right justified, and then the format (/FR) and the label (FY) were replicated together across the row by placing the cursor at B4 and typing

/R:C4.F4 RETURN

The Replicate command is explained in Chapter 3, Getting Started, and in the Replicate section of Chapter 4, Commands.

On row 5 the characters 1, 2, 3, 4, and 5 were entered as values, and as such are right justified by VisiCalc. Here VisiCalc values are used as report column headings. Occasionally this can simplify title entry, as here. For instance, we may want to enter years as labels; for example, we may want to use the years 1900, 1901, 1902, ... 1950 as titles. They can be conveniently entered as VisiCalc values.

The Repeating Label command was used to create the simulated underline at location A6, and it was then replicated across the row.

Finally the characters 1ST, 2ND, 3RD, and 4TH down column A had to be entered by first entering a double quote. This was necessary since the numbers 1, 2, 3, etc., if used as the first character of an entry, cause the entry to be a value and not a label. This means that we can't enter the letters after the numbers unless the double quote is used.

Using Labels as Documentation

Labels can be used within the sheets to provide helpful directions to users of the sheet, including ourselves, as we'll see in Chapter 9, Creating Templates, and Chapter 10, Documentation. For example, we can place the label

/PP E3 RETURN

onto row A starting at A1 (or any other location) by entering the double quote as necessary and dividing this into separate entries. In this example we've placed a Print command on the sheet as a reminder of the steps necessary to print a portion of the sheet. The full directions for printing are provided on this line. Additional examples are provided in the two chapters mentioned.

This line serves only as a reminder and can't be executed automatically by VisiCalc.

Evaluating Labels When Used in Formulas

Labels are evaluated as zero (0) if they are referenced in a formula. This can be either an advantage or disadvantage, depending on the usage. It's an advantage when used in electronic sheets

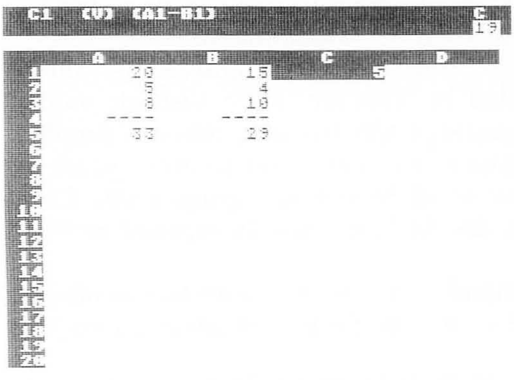


Figure 5-2. This figure shows our screen before we replicate the formula at C1 down column C.

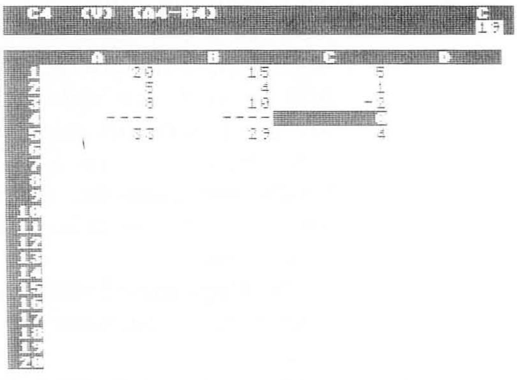


Figure 5-3. After replicating C1 from C2 through C5, we see that C4 has been given a value of zero because each underline is valued at zero, and their difference is therefore also zero.

in problems such as the examples in the @SUM section of Chapter 6, Built-in Functions. There we specify a range of a series including the element above and below the numbers to be totaled. This allows for rows to be added or deleted within the series without changing the formula. However, zero evaluation is a disadvantage in examples such as the replication of a formula involving subtraction of elements of one column from another. If some elements are blanks, or underlines, their difference will be zero.

This is demonstrated in Figure 5-2 and Figure 5-3, where we've placed some values down columns A and B with an underline at row 4. C1 contains the formula (A1-B1). When we replicate this formula down column C, we get the zero (0) shown at C4 in Figure 5-3, not an underline. This illustrates an additional reason for postponing the labeling process until other parts of the sheet have been completed. If row 4 is added after replicating, then we won't need to "clean up" the sheet as is now necessary with Figure 5-3.

INTRODUCING NUMBERS

Numeric values include data such as 1 or 3.14159 or −356, etc. Numbers in VisiCalc begin with one of the digits 0 to 9, with the positive or negative sign, or with a decimal point. Values can also begin with a left parenthesis, with the pound sign (#) to be discussed, or with the @ indicating a function.

Numbers are not allowed to have commas, dollar signs, or other nonnumeric characters (with the exception of E as will be explained).

Let's look at some valid and invalid numbers:

Valid Numbers
1000
−247.56
+197.000
0002

Invalid Numbers

Number	Reason why invalid
	1,100 accept the characters but evaluate them as ERROR.
$312.13	No $ allowed. VisiCalc will not allow you to start an entry with this character.
42.00CR	No letters allowed (except E as explained later).
TEN	No letters allowed; however, VisiCalc will accept this as a label and as such it will be evaluated as zero (0), not 10.

VisiCalc ignores extra zeros before the number, called leading zeros, and extra zeros to the right of the decimal point, called trailing zeros. If we enter 15.2 or 0015.2 or 15.20000, all are stored as 15.2.

Significant Digits: Values Stored versus Values Displayed

VisiCalc usually stores up to eleven significant digits for a value stored. Let's see what we mean with the following numbers.

		VisiCalc displays	
We type	VisiCalc stores	(Column width 9)	(Column width 18)
1	1	1	1
75	75	75	75
.0173	.0173	.0173	.0173
745.896	745.896	745.896	745.896

In each of these examples, VisiCalc acts as we expect. Values are stored and displayed as entered.

Now let's look at the same set of numbers typed and stored but displayed with various formats and a narrower column width.

		VisiCalc displays			
We type	VisiCalc stores	/GFD	/GFG	/GFI	/GF$
1	1	1	1	1	1.00
75	75	75	75	75	>>>>
.0173	.0173	.017	.017	0	0.02
745.896	745.896	746.	746.	746	>>>>

In these examples there are several things to notice. First, numbers are always displayed with at least one blank at the left of the column. This is done to prevent the confusion that would result if numbers from one column ran together with numbers in an adjacent column. Second, numbers are not always displayed the same way that they are stored. This is very significant.

Third, at times it may not be possible to display the value stored with the column width and format specified. For example, look at the stored value of 75, which we have tried to display with a column width of 5 and a dollar and cents format of two decimal places. This displayed value becomes 75.00 with that format. This has five columns; since VisiCalc leaves one blank at the left of the field, this cannot be displayed in a column five characters wide. When this occurs, VisiCalc fills the field with the greater-than symbol (>) except for one blank at the left. The other examples are explained in the Format command section of Chapter 4, Commands.

Let's look at additional examples.

We type	VisiCalc stores
987654321	987654321
4000000000000	4000000000000
123486123456789	123486123450000

When we look at the first two examples, it seems that VisiCalc can accurately store extremely large numbers. However, look at the third value. Here the value stored is different from that entered. VisiCalc is usually limited to storing eleven significant digits. The meaning of significant digits is illustrated with the third line. The decimal place has been accurately preserved, but digits after the eleventh digit (the 5) have been lost and zeros stored instead. The magnitude, or size, of the number has been roughly preserved but the accuracy has been sacrificed. For some electronic sheets, particularly in the scientific area, being close is sufficient. For others, it's insufficient. VisiCalc users need to recognize this limitation on accuracy and use caution as appropriate in preparing spreadsheets. We'll see additional examples in the next section.

Scientific Notation

As mentioned, scientific users often work with numbers of very large or very small magnitude. VisiCalc provides scientific notation, common in computer languages, to aid in storing and displaying large and small values. Let's look at some numbers written in scientific notation and explain their meaning.

Number	Means
1.4E4	In this "number," and in all numbers written in scientific notation, the letter E means, and is read as ____ times 10 raised to the ____ power where the blanks are replaced by the values to the left and right of the E. If we do that with the number shown here, we have the value 1.4 times 10 raised to the 4th power or 1.4 times 10000 or 14000. Therefore, 1.4E4 is equal to 14000.
1.4E+4	This is identical to the preceding example. Here the plus sign (+) is redundant.
1.4E−4	This is evaluated by raising 10 to a negative value, meaning that we are multiplying the initial value by a small number. We evaluate this as 1.4 times 10 raised to the −4th power or 1.4 times .0001 or .00014

From these few examples we have the concepts of scientific notation without realizing their real impact in terms of VisiCalc and other computer languages. Let's examine additional examples.

Number	Means
6.21597E20	When we convert this number from scientific notation to "normal" decimal notation, we have 6.21597 times 10 to the 20th power or 6.21597 times 100000000000000000000 or 621597000000000000000
6.21597E−20	Again we'll convert to decimal notation 6.21597 times 10 to the −20th power or 6.21597 times .00000000000000000001 or .0000000000000000000621597

From these two examples we can observe several things. First, it's possible to express numbers with extremely large or small magnitudes in a simple abbreviated notation. Second, the notation itself can require fewer characters to display and store than the full decimal equivalent. For example

6.21597E-20

requires twelve characters to display (including a blank at the left), while

.0000000000000000000621597

requires 27 characters to display.

It's for this reason that scientific notation is used by VisiCalc. VisiCalc will revert to scientific notation for display (and storage as we'll see) of values that could not otherwise be shown.

Look at the items in Figure 5-4 for some additional information about scientific notation and the magnitude of values permitted in VisiCalc. This spreadsheet was created by placing the value 1 into location A1 and then multiplying that value by 10 relatively down column A. Similarly the value 1 at C1 was multiplied by 0.1 relatively down column C.

At location A9 we see that VisiCalc begins to display the value in scientific notation, for it is at this location that the value cannot be fully displayed with at least one blank to the left.

The VisiCalc program makes this decision, for we cannot control this format directly. VisiCalc does not contain scientific notation among the various formats available for our use.

Notice in the lower portion of the screen that we begin to display an ERROR value in column A between 1E61 and 1E62 and

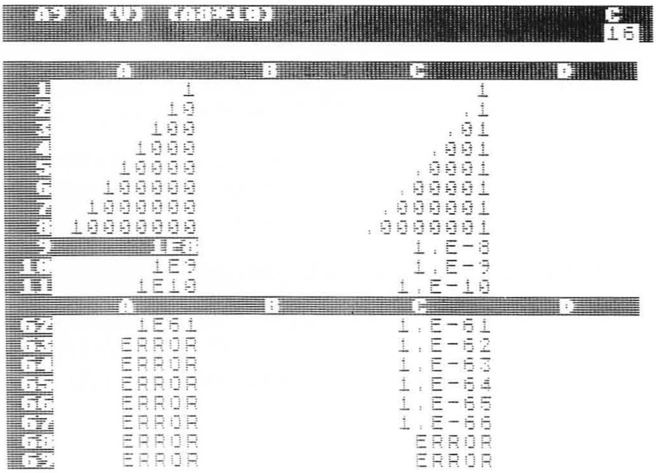


Figure 5-4. This demonstrates in scientific notation, the magnitude of the largest and smallest numbers that can be stored with VisiCalc.

an ERROR value in column C between 1.E-66 and 1.E-67. These values thus provide a relative range (and not exact values) of the largest and smallest numbers that VisiCalc can store.

We can also enter values in scientific notation, with some limitations. Following are some acceptable values that we can type as shown and, which will place the value shown in the entry. The first item, 12E30, will be accepted and then stored as 1.2E31.

12E30
9.8734E-15
1.8E41

However, we should not enter items such as E20 or E-12 or +E61 since the first two will be accepted as a LABEL (the first character of each is a letter), and the third will be accepted as a formula, with E61 an entry coordinate rather than scientific notation.

A final important concept in working with this notation is the combination of the magnitude of the number with the number of significant digits stored by VisiCalc. Suppose we type

1234567890123456789

as the entry value at a location. When we press RETURN, the entry line shows

1.2345678901E18

which is the way the value is stored. Let's closely compare these values and subtract the stored from the entered value:

entered:	1234567890123456789
minus stored:	1234567890100000000
difference:	23456789

For these values the difference illustrates the significant compromise that we must accept with scientific notation in VisiCalc. We give up accuracy beyond 11 significant digits in order to obtain a value that is approximately correct when considered relative to the magnitude of the values involved.

In this example, with the global column width set at 9, this value will be displayed as

1.235E18

illustrating that VisiCalc will display the value as closely as possible to the stored value within the specified column width and format.

Using the Pound Sign (#)

The pound sign (#) serves several related purposes in revealing or including a value in an entry. Let's start with the example of finding the circumference of a circle where we want to have the ability to change the radius easily and obtain the circumference. Suppose that we've prepared the electronic sheet of Figure 5-5.

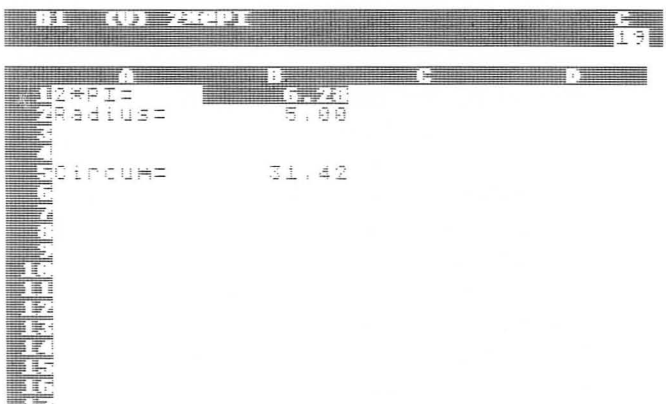


Figure 5-5. This sheet will be used to demonstrate the use of the # sign.

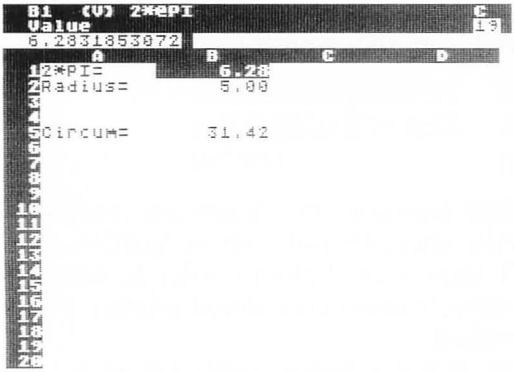


Figure 5-6. Using the # key places the current values of this entry on the edit line.

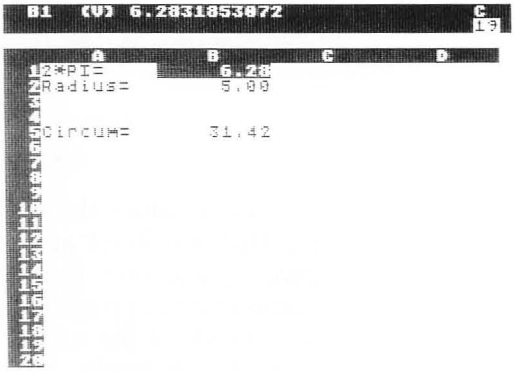


Figure 5-7. The # key followed by a RETURN replaces the contents with its current value.

With the cursor at location B1, we see the formula that we entered earlier as

2*@PI

At this point if we press the # symbol, we get the edit line shown in Figure 5-6. On this screen we are able to see the value of the formula with the full number of significant digits. If we press BREAK, we're back to where we were; however, if we press the RETURN key, we obtain the changes shown in Figure 5-7. The formula has been replaced by its value. This is called pounding the value into place.

Let's look at another example on the same sheet at location B5, where we have the results of the formula (B1*B2) displayed using the dollars and cents format to display two decimal places. Suppose that we need to see more than two decimal places for the radius at B2 but don't want to change the column width or format. We can instead press the # key, get our value shown on the edit line with all significant digits stored, and then press BREAK when we're done, without changing the sheet.

Let's look at another usage of the # key. As we are entering a formula, we can use this key immediately after entering a coordinate in order to have the value at the location of that coordinate permanently embedded in the formula instead of the coordinate. For example, at location B5, when we are entering, or editing the formula, if we enter

(B1#

this immediately leaves the following on the edit line:

(6.2831853072

followed by the box. If we then type

*B2) RETURN

we have completed our formula at B5.

Using the Exclamation Point (!)

If, when entering a formula, and before pressing RETURN or an arrow key, we type the exclamation point (!), whatever formula has been entered on the line so far will be replaced by its numeric value. For example, with the same sheet as in Figure 5-5, if we enter

(B1*B2)!

at location B5, the formula is instantly replaced by its current numeric value, here 31.415926536. This is followed by the box on the edit line waiting for us either to append additional formula or to press RETURN or an arrow key.

FORMULAS

Formulas usually involve a relationship between items where the items may be numbers, coordinate references, or built-in functions. Some examples follow

Formula	Comments
18000+21000	This is the simple relationship of addition between two constant numbers. We might type such a formula if we're using VisiCalc as a calculator or if we're entering values from several budget worksheets and want to preserve the separate numbers as well as use their sum.
12+B3	Our formula adds 12 and the value at B3 and places the result at the location of this formula.
B3+12	Although this is mathematically equivalent to the previous example, it is not equivalent in VisiCalc. Because this starts with a letter, in VisiCalc it

	becomes a label that will be given the value zero (0) if referenced in another formula.
+B3+12	This usage is a formula because the plus sign (+) initiates a value, not a label.
(B3+12	This is also a formula with the same value as the previous example. A left parenthesis can be used to begin a formula. VisiCalc doesn't require a closing parenthesis, commonly required in most computer languages. For the experienced user, concerned with economy of keystrokes, this is a timesaver. However, for the neophyte computer user, caution is necessary, as we'll see, and closing all open parentheses may be a valuable habit to develop.
(B3+12)	This is numerically equivalent to the previous value. Here we've added a right parenthesis.

In the last three examples we see three ways to enter the same formula. Which is best? As suggested, that may vary, depending upon our skill level. What is important is that we choose one method and use it regularly.

Let's look at additional formulas.

Formula	Comments
@SUM(A1...A8)	A built-in function is a formula.
(230+2*@PI)	This formula also uses a built-in function.
@PI*R1∧2	This formula demonstrates the use of the exponentiation symbol (∧). In VisiCalc five mathematical operators are permitted. They are

Operator	Indicates
+	Addition
−	Subtraction
/	Division
*	Multiplication
∧	Exponentiation (raise to a power)

However, this example raises a more important question. When more than one operation occurs in a formula, which is evaluated first? This topic, called the order of computation, is discussed next.

VisiCalc's Order of Computation

That last formula above looks like the familiar formula for the area of a circle; however, it's not in VisiCalc. VisiCalc, in the absence of parentheses, performs computations from left to right. This is very important to remember. It means that the formula is evaluated by multiplying @PI times R1 first and then raising the product to the second power. How can we get what we want? There are several options. We can write

R1∧ 2*@PI

which causes the exponentiation to occur first as we want.

We can also write

@PI*(R1∧ 2)

because VisiCalc will perform calculations within parentheses first.

Let's look at additional examples. In each case the underline indicates the current computation.

Formula	How evaluated
$\underline{16+4}/5$	$= \underline{20/5} = 4$
$16+(\underline{4/5})$	$= \underline{16+0.8} = 16.8$
$\underline{21+14}/7*2*6$	$= \underline{35/7}*2*6 = \underline{5*2}*6 = \underline{10*6} = 60$
$21+(\underline{14/7}*2)*6$	$= 21+(\underline{2*2})*6 = \underline{21+4}*6 = \underline{25*6} = 150$
$21+(14/(\underline{7*2}))*6$	$= 21+(\underline{14/14})*6 = \underline{21+1}*6 = \underline{22*6} = 132$

In the last example we see that sets of computations "nested" within parentheses (that is, one within another) are evaluated from the inside out.

Efficiencies in Preparing Formulas

Although many electronic sheets require merely seconds, or less, to recalculate an entire sheet, some sheets may require a significant amount of time. For some sheets the manual recalculation initiated by an exclamation mark (!) and discussed under the Global command in Chapter 4, Commands, will minimize overall waiting time. However, for other sheets that require regular recalculation we may find ourselves with a longer wait for the sheet to be recalculated.

Let's look at an example in which the way that we write a formula influences the length of time necessary to calculate all entries of the sheet. In Figure 5-8 a formula commonly used for tests of this kind has been entered. It is

$$A(X)^3 + B(X)^2 + C(X) + D$$

The constant values of A, B, C, and D in the formula are 12, 6, 3, and 5. The value of X is stored in entry A1. We'll place this formula at A2 and replicate it through A200. Then we'll change the value at A1 and record the time necessary for the full calculation of the sheet.

If we enter the formula without parentheses then VisiCalc works from left to right and does not give us the value that we wish. Therefore, we'll parenthesize carefully and place the formula

12*(A1^3)+(6*(A1^2))+(3*A1)+5

in locations A2 through A200. When the value at A1 is changed from 0 to 10, the full recalculation of the sheet requires approximately 220 seconds (almost 4 minutes).

Figure 5-8. A sheet where the manner in which we enter the formula affects the recalculation time.

We'll rewrite the formula, without exponentiation, and try again. Our formula, arithmetically identical, will now be written as

12*A1*A1*A1+(6*A1*A1)+(3*A1)+5

This revision reduces the time required from over 200 seconds to about 7 seconds.

Finally, let's try with a third version of the same formula. This time we'll write

(((((A1*12)+6)*A1)+3)*A1)+5

at A2 through A200. This recalculation required 5 seconds.

The purpose of these examples is to demonstrate that the way in which we write formulas can significantly affect timing on some sheets. On most sheets involving addition, subtraction, multiplication, and division, timing may not be a problem. But on sheets with many built-in functions or with exponentiation, we can sometimes reduce the time required, as shown.

Let's list some efficiencies that can reduce calculation time. Using these suggestions may make sense only if we are working on a sheet that is used often and recalculated often. That is, use the suggestions if the time that we save by following them is greater than the time required to implement them.

- If one computed value is used at many locations, compute it one time at one location (not repeatedly at each occurrence) and refer to its value.
- Trigonometric functions are slow. If we calculate the same function for an angle in many places, establish a single entry in which that function is calculated and reference it elsewhere as needed.
- Pound constant values into place instead of repeating them (as illustrated previously with 2*@PI).
- Exponentiation is a slow operation; avoid it if repeated multiplication can be used.

In general if we have several ways to perform a calculation, we can establish a worksheet like that used in Figure 5-8 and actually time each method to determine which is faster.

Chapter 6

Built-In Functions

INTRODUCTION

This chapter discusses the built-in functions provided by VisiCalc. Most of these functions perform a numerical computation for us, using values that we supply or reference. For example, there is a built-in function to find the sum of specified entries, another to find the average of selected values, others to find the sine, the cosine, or the tangent of angles, others for logarithms, etc. In this chapter, we'll discuss each of these in alphabetic order by the name of the function.

Some general considerations apply to all functions:

- We initiate a function by typing the character @. This is the beginning of a function; as the first character of a line, it indicates that we'll have a value here.
- Functions may appear other than at the beginning of a line, and more than one function may be used in a formula.
- Following the @ we type the name of the function, for example,

@SUM	to begin the sum function
@SIN	to begin the trigonometric sine function.

When typing the name of the function, it is possible with VisiCalc to type only part of the name followed by the left parenthesis, and VisiCalc will complete the function name. For example, if we type

@T(

this becomes

@TAN(

on the edit line. This should be used carefully since typing

@A(

will become

@AVERAGE(

when we may have wanted

@ABS(or @ACOS(or @ASIN(or @ATAN(

and not

@AVERAGE(

To ensure that we do obtain the average function, here we must at least type

@AV(

Similarly,

@AVE(and @AVER(and @AVERA(and @AVERAG(

all will result in placing the average function on the edit line as well as the full name, @AVERAGE(.

- The left parenthesis terminates naming the function and indicates that we are ready to enter the function "arguments" if required (some functions have no arguments).
- The arguments of a function are the values on which we want the function to operate. Arguments may have several forms. For example, we can write

Function	Explanation
@SUM(A1,A12,B13)	This will provide the sum of the three items listed. Here we have three arguments separated by commas.
@SUM(A1...A8)	This will calculate the sum of all entries from A1 through A8. As with other series, when one period (.) is typed

	VisiCalc generates the three periods of the ellipsis. This example has one argument, here a series. Even though there are eight entries here, there is only a single argument.
@SUM(A1...A8,A12,A16)	Here three arguments occur, the series A1...A8, then A12, and then A16. Again the sum is computed. Not all functions can have more than one argument. For example, we can only request the sine for a single argument and that must be one value, not a series. This will be discussed for each function described next.
@SUM(A1...A8,@SIN(A32))	This demonstrates the use of one function as an argument of another function.
(A7-@SUM(A18...A30))	A function can also be used as part of a formula or expression, as shown here.
@SUM(A1...A8,A12*3)	The second argument in this example is an expression, the product of A12 and 3.

• In many cases the entire function, from the initial @ to the final) becomes one numeric value. This is not true for the @NA or @ERROR functions, as will be explained.

• The arguments, if present, and the function itself are terminated by a right parenthesis) except for those functions without arguments. Although the right parenthesis is not required, its use can avoid confusion.

In the following discussion we'll describe each built-in function separately and, where appropriate, suggest possible uses of the function. Where the term "argument" or the terms "argument1, argument2, ... " appear, substitute values or references as discussed.

@ABS(ARGUMENT) THE ABSOLUTE VALUE

The absolute value of a number is its numeric value, ignoring the sign. For example, the absolute value of 5 is 5, of −5 is 5 (the sign is ignored), and of 0 is 0. The argument must evaluate to a single value. For example, an argument like B3 or B3*(A12-C14) is acceptable, while arguments of A16...A30 or A7,A12,3*A18 are not and will be evaluated as ERROR.

Refer to the @ERROR section of this chapter for a discussion of the ERROR value.

@ACOS(ARGUMENT) THE ARCCOSINE

The arccosine is the angle, in radians, with a cosine equal to the value of the argument. In requesting the arccosine we are asking for the angle whose cosine has the value that we provide as the argument. Refer to angle A in Figure 6-1; the cosine of angle A is the ratio of the length of side b to side c, that is, the COS $A = b/c$. This ratio will always have a value between −1 and +1, and in the function an ERROR will result in the field if the argument is not evaluated within this range. The argument must evaluate as a single numeric value and can't be a series because that will result in a value of ERROR.

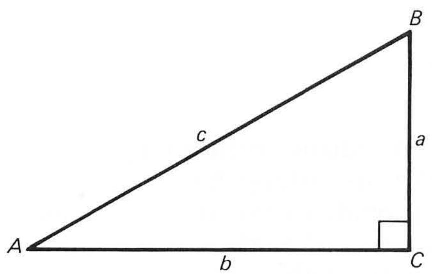


Figure 6-1. A triangle that will be referenced in discussions of the trigonometric functions.

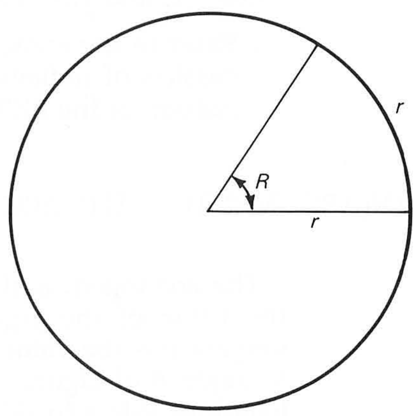


Figure 6-2. A visual representation of one radian, the angle R, formed at the center of a circle by an arc the length of the radius.

A radian is the angle formed at the center of a circle, subtending an arc on the circle equal to the length of a radius. In Figure 6-2, the radius (r) has been measured along the circumference of the circle, and the resultant angle R at the center is one radian. Since the total length of the circumferance is the product of $2\pi r$ this means that there are 2π radians in the circle of 360 degrees. Therefore one radian is approximately 57.2958 degrees (360 degrees divided by 2π). This explanation is intended only as a very brief introduction to these topics.

With this function, angles are returned as values from 0 (for a cosine of 1) to π (for a cosine of -1).

Refer to the @ERROR section of this chapter for a discussion of the ERROR value.

@ASIN(ARGUMENT) THE ARCSINE

The arcsine is the angle, in radians, with a sine equal to the value of the argument. We are asking for an angle whose sine has the value that we provide as the argument. Referring to angle A of Figure 6-1, the sine of angle A is the ratio of the length of side a to side c, that is the SIN $A = a/c$. This ratio will always have a value between -1 and $+1$, and the value ERROR will result in the entry if the argument does not evaluate within this range. The angle returned will range between -1.5707963268 radians ($-90°$) for a value -1, and 1.5707963268 ($+90°$) for a value of $+1$.

Refer to the @ACOS section of this chapter for a brief discussion of radians, and to the @ERROR section for a discussion of the ERROR value.

@ATAN(ARGUMENT) THE ARCTANGENT

The arctangent is the angle, in radians, with a tangent equal to the value of the argument. We are asking for an angle whose tangent has the value that we provide as the argument. Referring to angle A of Figure 6-1, the tangent of angle A is the ratio of the length of side a to side b, that is the TAN $A = a/b$. Since there are no numeric limits on this ratio, the limitation becomes the capability of VisiCalc. Here the arctangent of $-1E61$ is -1.5707963268 radians ($-90°$) and the arctangent of $+1E61$ is $+1.5707963268$ ($+90°$).

Arguments of −1E62 or smaller, or of 1E62 or larger, result in an ERROR.

Refer to the @ACOS section of this chapter for a brief discussion of radians, and also to the @COS and @ERROR sections. Refer to Chapter 5, Labels, Numbers, and Formulas, for a discussion of scientific notation (that is, to review the meaning of the values −1E61 and +1E61).

@AVERAGE(ARGUMENT1,ARGUMENT2,...) THE AVERAGE

The average is the single value that results when quantities are summed and that sum is then divided by the number of quantities. For an example of the functioning of @AVERAGE, refer to Figure 6-3 in which we have recorded test scores for several students and used the @AVERAGE function to compute their averages. Let's look at each carefully to understand fully how blank entries affect the computations:

Student	Explanation of the @AVERAGE function
ALICE	The function at location B10 is @AVERAGE(B5...B8) and the average is 90 as expected.
BOB	The same function is used here as with ALICE, and the result is 80, the sum 80+80+80 (or 240) divided by 3, giving 80. The last entry C8, which is blank, is ignored in the count of the items in the column. This is desirable if we are waiting for BOB to take the last examination and want to have his average "to date" for the three exams taken; however, if Bob did not take the last test, and the blank entry really means that he has received a 0, then we will have to enter either a 0, which will produce an average of 60 at location C10, or use the function differently as discussed for CAROL.
CAROL	The function used here and shown in Figure 6-4, is

@AVERAGE(D5,D6,D7,D8)

which produces an average of 60. Here every argument is counted as an item whether it is blank or not. If this is not desired, then the formulation shown for BOB could be used, or a series such as

@AVERAGE(D5...D7,D8...D8)

could be used.

The @AVERAGE function must be used with care if blank entries are expected in any of the references of the arguments. Since the @AVERAGE function is equivalent to the @SUM function divided by the @COUNT function, a thorough understanding of how both of them function is necessary, especially since the @COUNT function handles blank entries in the unique way described here.

Refer to the @COUNT and @SUM sections of this chapter for additional relevant discussion.

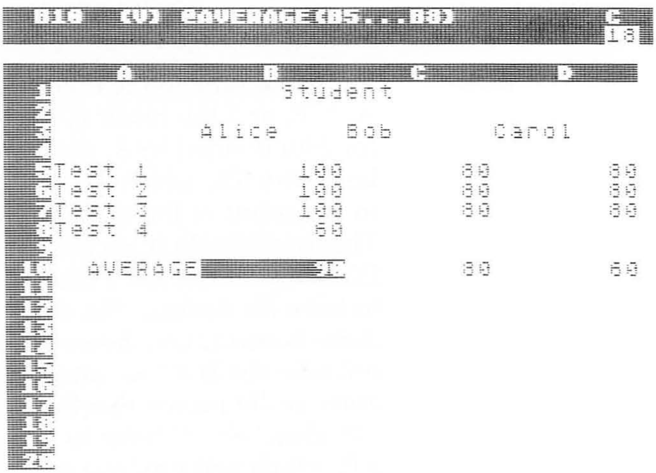


Figure 6-3. Examples of the effect of blank entries when used with the @AVERAGE function.

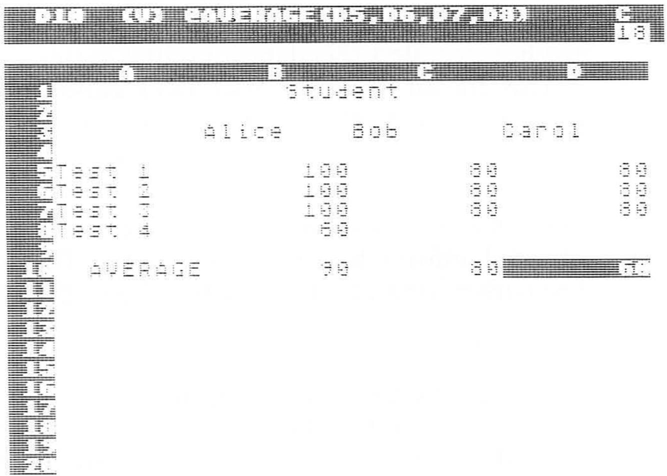


Figure 6-4. Another usage of the @AVERAGE function to count blank entries.

@COS(ARGUMENT) THE COSINE

The cosine of an angle, for example, angle *A* of Figure 6-1, is the ratio of the length of side *b* to the hypotenuse, side *c*; that is, COS $A = b/c$. In this function the argument is an angle provided in radians. The cosine is returned with the appropriate sign (positive or negative), depending on the quadrant in which the angle falls.

Refer to the @ACOS section of this chapter for a brief discussion of radians.

@COUNT(ARGUMENT1,ARGUMENT2,...) COUNT HOW MANY

The @COUNT function provides a single value, the number of nonblank entries in the arguments referenced (with one exception as will be noted).

For example, Figure 6-5 contains information received from the reader response cards common in magazines. Suppose this service limits people to choosing up to five items about which they wish to receive product information. Here TOM has asked for information on advertisements 32, 58, and 104. We'll assume the addresses and actual mailing appear in a different system and that we have created this data from that system.

This figure uses the @COUNT function in column I to count how many requests each person has made. The formula at I5 is

@COUNT(C5...G5)

and results in a value of 3. This indicates that the person has requested information for three items. The highlighted formula in the cursor location of Figure 6-5 shows the formula

@COUNT(C8...G8)

and gives the value 0 as expected at location I8.

However, look at Figure 6-6, and in particular at the value 5 in the cursor. An often undesirable feature of the @COUNT function is that a single entry included as an argument will always be counted as 1 (one) whether the entry is blank or not. Thus, the function here at location I9 of Figure 6-6 contains

@COUNT(C9,D9,E9,F9,G9)

which will always count each of these as 1 no matter what values they contain. When it is necessary to include a single value and the results discussed are not desired, then the entry must be listed as a series, for example

(D9...D9)

which is an awkward usage.

This is a limited function. In our example we cannot now use it to count how many of the people requested one mailing, how many requested two, etc. The function only counts nonblank entries. A zero entry will be counted as 1, which we can see in the row for TONY in Figure 6-6.

This function, although it only counts nonblank entries, can also be used to obtain the number of blank entries by using it in a formula that subtracts the value counted from the total entries. For example, in another problem, if we wanted to know the number of blank entries in the 15 entries from G1 through G15, we could obtain this value with the formula

15-@COUNT(G1...G15)

The functioning of the @COUNT is equivalent to the computation of the denominator in the @AVERAGE function.

Refer also to the @AVERAGE section of this chapter.

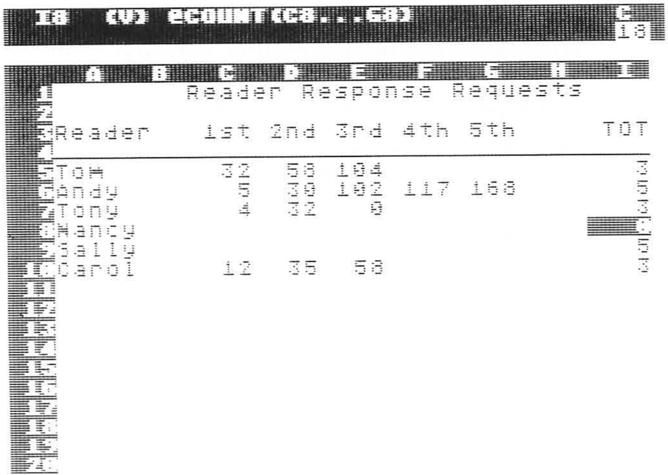


Figure 6-5. An example of the @COUNT function.

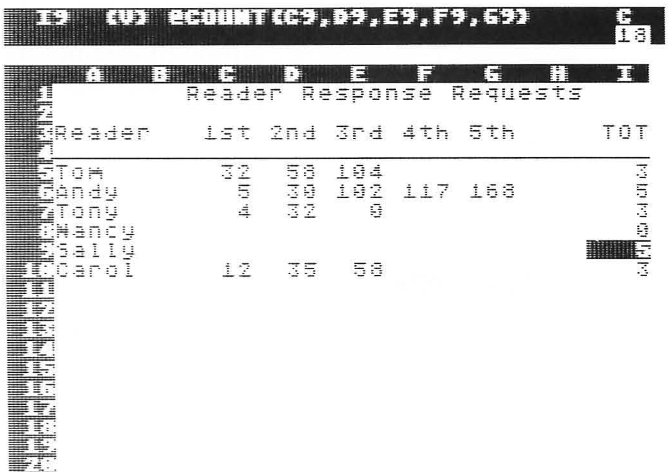


Figure 6-6. An additional example of the @COUNT function where a "wrong" result occurs at location I9.

@ERROR THE ERROR FUNCTION AND ERROR VALUE

This function requires no argument and is used by deliberately typing

@ERROR

This function may appear wherever functions are permitted (i.e. within a formula, within another function, in an expression, alone).

An entry containing this function will have the *value* ERROR (as opposed to the *function* @ERROR).

The value ERROR can occur at an entry for many different reasons. When it does, all other entries referencing it will have value ERROR. For example,

Entry contains	Explanation
16*@ERROR/81	This formula will be evaluated as ERROR. The formula is not destroyed, but the value is displayed as ERROR on the sheet.
@ERROR	The same results are obtained as in the preceding formula, that is, the value ERROR is displayed here.
@SUM(A1...A3...A12)	This function will be accepted, although it is not in a correct format, but will be evaluated as ERROR.
@COS(B1,B2)	The cosine function requires one argument, not two. VisiCalc will accept two but evaluate them as the value ERROR.
@TAN(1.5707963268)	This function, the tangent of 90 degrees, will generate an ERROR value, as will other trignometric functions when the function has no value at the given argument.
(A1/C3)	If C3 is evaluated as zero, then the attempt to divide by zero will place the value ERROR at this entry.
@AVERAGE(A1...A4)	If all entries from A1 to A4 are blank, then the average is computed as ERROR. This is true since this function only counts nonblank entries, and the number of entries will therefore be zero, resulting in an attempted division by zero.
@SQRT(B3)	If B3 is a negative value, the square root (refer to the @SQRT function

later in this chapter) will be the value ERROR.

These examples are representative of the type of activity that will display the value ERROR on our sheets. This can be a common occurrence in creating templates (refer to Chapter 9, Creating Templates) when we have entries that either are blank or contain labels while awaiting data entry activity.

@EXP(ARGUMENT) *E* TO A POWER

This function provides the value of the mathematical constant *e* raised to the power of the included argument. If a value for *e* itself is required, using the function with an argument of 1 will provide a value of 2.718281828 for *e*.

@INT(ARGUMENT) INTEGER

This function removes any fractional part of the argument, leaving only the integer portion. Let's look at some examples:

When	And A1 contains	
B1 contains	Formula	Evaluates as
6	@INT(B1)	6
0	@INT(B1)	0
−3	@INT(B1)	−3
6.4	@INT(B1)	6
6.8	@INT(B1)	6
−3.4	@INT(B1)	−3
−3.8	@INT(B1)	−3

In these examples, all entries at A1 not only are displayed as integer values but also are stored as integers.

This operates differently from the integer formats (/FI or /GFI) in several ways. First, the integer format (not the @INT function) affects only the display on the screen and not the entry value. Second, the displayed value with the integer format will be rounded; this is different from the action of the @INT function. With the integer format (not function) the following occurs:

Entry value at B1	Format at B1	Screen display of B1
6.4	/FI	6
6.8	/FI	7
−3.4	/FI	−3
−3.8	/FI	−4

These two capabilities must be used with care to ensure that desired results are achieved. For example, with a global integer format it's possible to have the following occur:

	1
	1
	1
TOTAL	4

What's happened is that the three individual entries listed as 1 above are actually each 1.3. Their total, which is 3.9, becomes the integer 4 in the TOTAL entry. To avoid this, if desired, each entry forming a part of the total should be computed within the integer function @INT.

Let's demonstrate this with a dollar and cents example where exact totaling is usually very important. Here we may not wish to display a total that is not the apparent column total. Look at the column at the right:

Entry coordinate	Entry value	Value displayed /GF$
A3	1.3333...	1.33
A4	1.3333...	1.33
A5 (total)	1.6666...	1.67

When this is unacceptable it is possible to ignore decimal places of each entry, without rounding, beyond two decimal places by multiplying the entry contents by 100 (effectively shifting the decimal place two places to the right), taking the integer value of the result with the @INT function, then dividing the results by 100 (to shift the decimal place back). A formula to do this is

@INT(*entry**100)/100

where *entry* is the value or formula at the entry.

If rounding of entries is desired (rather than ignoring the decimal places to the right of two places), then the formula

@INT((*entry*+0.005)*100)/100

can be used. This provides a value rounded to two decimal places. This should be done for each separate entry before totaling and will then produce a correct total. If this formula is used at entries A3 and A4 in the preceding example, we have

Entry coordinate	Entry value	Value displayed with /GF$
A3	1.33	1.33
A4	1.33	1.33
A5 (total)	1.66	1.66

We'll need to decide when this is necessary.

@LN(ARGUMENT) NATURAL LOGARITHM

This function returns the natural logarithm of the argument, that is, the value to which *e* must be raised to produce the argument. A single positive argument will return a value; multiple arguments, for example A1...A4, A8, or arguments that evaluate to zero or to a negative number produce a value of ERROR.

Refer to the @ERROR and @EXP sections of this chapter.

@LOG10(ARGUMENT) LOGARITHM, BASE 10

This function operates similarly to the @LN function just described with the exception that the base is 10, not *e*. The third character of this entry is the letter O and the sixth character is the digit zero.

@LOOKUP(ARGUMENT1,ARGUMENT2) LOOK UP A VALUE IN A TABLE

This function performs a simple search, or lookup, of a value that we provide (argument1) in a table that we place on the sheet (argument2). The example of Figure 6-7 illustrates how this

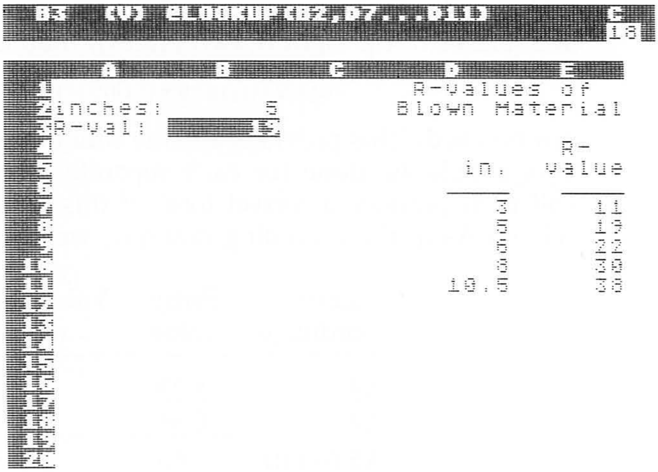


Figure 6-7. The use of the @LOOKUP function in an energy-related example.

operates. This sheet has been set up with a simple energy-related table containing the R-values of insulation blown in at depths varying from 3 inches, at which the R-value is 11, to 10.5 inches, at which the R-value is 38. For our problem we have used location B2 to hold the number of inches for which we want to look up the R-value in the table of columns D and E. The R-value that is found will be placed in entry B3 in which we have placed

@LOOKUP(B2,D7...D11)

> This can be read: Look up the value currently in location B2 in column D from D7 through D11. When there is a match, put the corresponding relative entry from column E into location B3 (the entry with the LOOKUP function).

Argument2 may be either a column series or a row series, and the entries corresponding to these must be in the column to the right, or the row below, argument2.

The values in argument2 should be sorted in ascending order, or incorrect values may result.

It's important to understand what occurs when argument1 is not in the table. The following examples clarify this, again using Figure 6-7. Let's place other values at B2, examine the results in B3, and explain what occurs.

B2 contains	Results at B3	Explanation
−2	NA	Not available. The lookup begins with the first entry of the series of argument2. If that entry is greater than argument1, the function returns the value NA.
0	NA	Same as the preceding −2.
3	11	The item is found, with the function returning 11.
7	22	The table (D7...D11) is searched. After proceeding to an entry where a value larger than argument1 is encountered, the function is given the value corresponding to the previous entry of the table. Here, as value 8 is reached, (8 being larger than 7), the returned value is 22, which is the value associated with the next smaller entry of the table.
10	30	Same as the preceding 7.
10.5	38	The item is found, returning 38.
15	38	Same as the preceding 7.

Also refer to the @NA section of this chapter.

@MAX(ARGUMENT1,ARGUMENT2,...) THE MAXIMUM VALUE

@MIN(ARGUMENT1,ARGUMENT2,...) THE MINIMUM VALUE

We'll consider these two functions together. The maximum and minimum functions, @MAX and @MIN respectively, accept arguments as just shown and return one value.

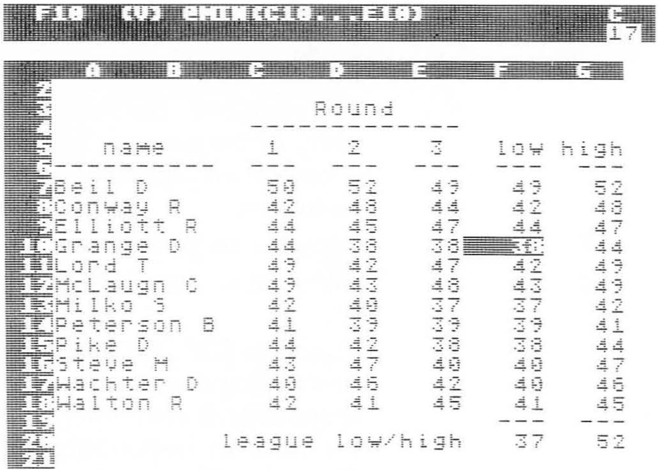
Figure 6-8 contains the golf scores of the members of a league for the three rounds played to date. Notice that column F contains the individual best score to date (the low score) and column G contains the individual worst score to date. In the LOW column we have placed a formula that states

@MIN(C10...E10)

for player GRANGE and in the high column for him we've placed

@MAX(C10...E10)

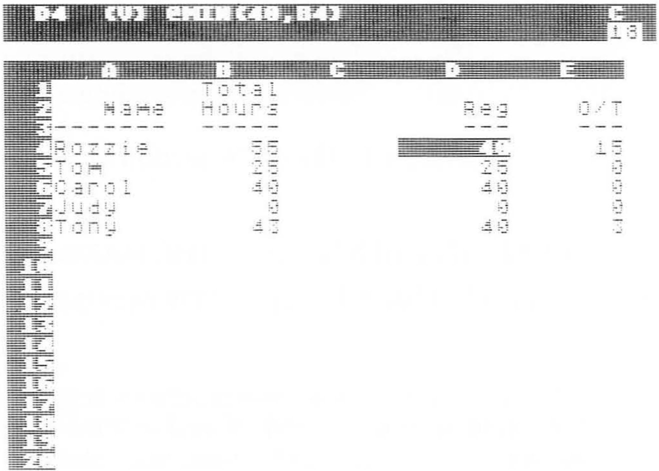
His best and worst scores are provided by these functions.



F10 (V) @MIN(C10...E10) C 17

		Round				
name		1	2	3	low	high
Beil D		50	52	49	49	52
Conway R		42	48	44	42	48
Elliott R		44	45	47	44	47
Grange D		44	38	38	38	44
Lord T		49	42	47	42	49
McLaugn C		49	43	48	43	49
Milko S		42	40	37	37	42
Peterson B		41	39	39	39	41
Pike D		44	42	38	38	44
Steve M		43	47	40	40	47
Wachter D		40	46	42	40	46
Walton R		42	41	45	41	45
		league	low/high		37	52

Figure 6-8. Examples of the use of the @MAX and @MIN built-in functions.



D4 (V) @MIN(40,B4) C 18

Name	Total Hours		Reg	O/T
Rozzie	55		40	15
Tom	25		25	0
Carol	40		40	0
Judy	0		0	0
Tony	43		40	3

Figure 6-9. Additional examples of the capabilities of the @MAX and @MIN built-in functions.

In the same figure notice the bottom of columns F and G where the formulas

@MIN(F7...F18) and @MAX(G7...G18)

have been written. These are similar to the usage in the rows for each player.

These two functions are simple in concept but can be powerful in a wide range of applications. For example, we can find the high and low test score of students in a class. High and low sales values can be selected. Even beyond these uses, a series of additional problems can be solved with this function.

For example, suppose that we have the total number of hours that an individual worked during the week and we want to determine how many hours were regular time and how many were overtime. A solution to this problem is shown in Figure 6-9. Data are entered for each person's name and total hours worked. Then formulas are used to determine the number of regular hours (REG) and overtime hours (O/T) worked.

The formulas in columns D and E follow the example presented next for the entries for ROZZIE at location D4 and E4. They are as follows

Location	Formula	Explanation
D4	@MIN(40,B4)	The value of the function will be the lower of either 40 or B4. Thus, if the person worked over 40 hours, as ROZZIE did, we display 40, the regular hours.
D5	@MIN(40,B5)	For an individual such as TOM, who worked a total of 25 hours, the function is evaluated as the lower of 40 or 25, here 25.
E4	(B4-D4)	Here if the total hours is over 40, as with ROZZIE, then this formula is evaluated as (55−40), which is 15. Therefore, we list ROZZIE's overtime hours correctly.
E5	(B5-D5)	Now for Tom the formula becomes (25−25), which is 0. Again we have the correct overtime.

An argument with an entry coordinate that refers to an entry that is blank or contains a label will be evaluated as zero.

Refer to the section "Detecting and Preventing Errors" in Chapter 8, Recognizing, Preventing, and Correcting Errors, where an additional similar use of the @MAX function is presented.

@NA NOT AVAILABLE

The @NA function, which requires no arguments, stands for not available, and returns a constant, like the built-in function @PI. When it is written into an entry of the sheet several things occur. First, the value NA (without the function symbol @) appears in the entry. Second, all entries that reference this entry are also calculated at the value NA.

This is useful in instances where we do not have values available and where if we left blanks, or the value 0, we would have undesirable results on the sheet. An example is shown in Figure 6-10 and Figure 6-11. In Figure 6-10 we have columns for the MONTH, QUOTA, ACTUAL SALES, and the DIFFERENCE (ACTUAL SALES minus QUOTA). Suppose that we are updating this sheet monthly as we receive sales figures. It is now JUNE. In the figure some of the ACTUAL SALES values have been left blank,

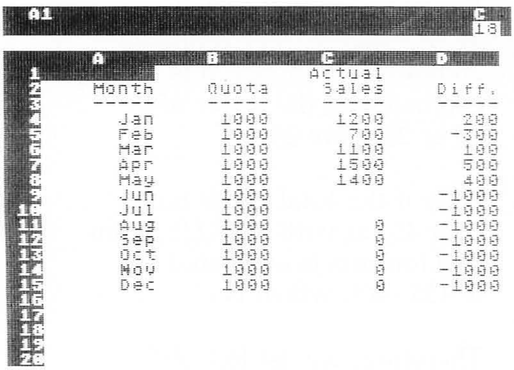


Figure 6-10. A sheet before using the @NA function. Misleading information is displayed in entries D9 through D15.

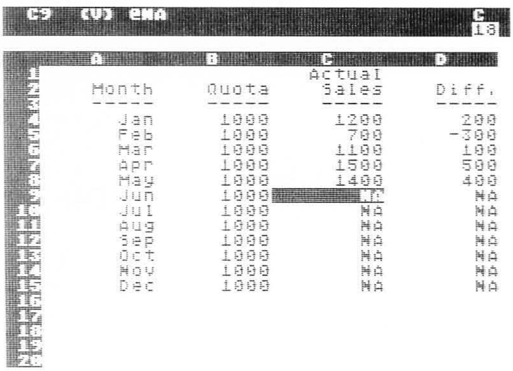


Figure 6-11. After entering @NA in entries C9 through C15, we have replaced the misleading information in column D of Figure 6-10 with the value NA, for not available.

and for illustration some of them have been valued at 0. Notice that the difference in both cases is shown as −1000. This is misleading since the sales figures for the month are not available yet.

In Figure 6-11 the ACTUAL SALES for JUNE through DECEMBER have been changed and entered as @NA. We do not change the formulas in column D. Notice that all the differences now also show the value NA. As we enter a numeric value for each month, our difference will be computed for that month.

The cursor at location C9 displays NA as the value, and the entry contents line shows @NA, the built-in function.

The @NA value can be useful on occasion to determine every entry on the sheet that refers to a specific entry. With no NAs on the sheet, place @NA at the desired entry, and print the full sheet; a visual cross reference of this particular entry appears.

The effect of this value is described in the discussions of many other functions.

@NPV(ARGUMENT1,ARGUMENT2) THE NET PRESENT VALUE

This function gives us the value today of money that we will receive. Let's look at an example that will explain the elements of this concept. Suppose that today we invest $100 at a prevailing simple interest rate of 15% for a year. At the end of the year we will have $115. Now if we ask what is the value today of $115 one year from now when the current interest rate is 15%, the answer is $100. To have $115 in a year, we must invest $100 now.

With this function, argument1 must be a single value, which is the interest rate, here called the discount, and entered as a decimal (that is 15% is entered as .15). Argument2 must be a series, the money available at the end of each time period.

Look at Figure 6-12 where two rows are entered as examples of the @NPV function. At location A1

@NPV(B1,C1...C1)

has been entered. This is the example with which we began this discussion. We're asking for the net present value of $115 one year from now if the interest rate is 15%. Our result, displayed at A1, is $100, as expected. (The sheet has a global format of dollars and cents.)

Look at row 4. Here we have the formula

@NPV(B4,C4...E4)

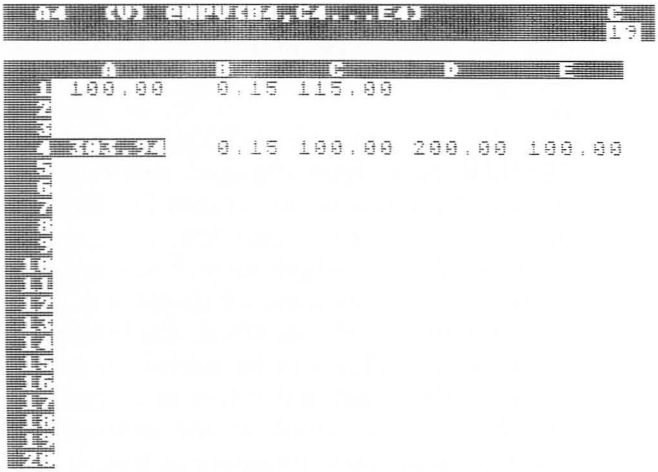


Figure 6-12. Two examples of the @NPV function.

which is asking what is the value today at 15% of $100 in one year, plus $200 in two years, plus $100 in three years. Our @NPV function says that it is worth $303.94 today.

Let's verify that result by working forward from the $303.94 as shown next.

Amount	Explanation
$303.94	We begin today with this amount.
+45.59	At the end of one year, our money has earned interest at 15%, returning us $45.59.
$349.53	The total of our principal plus interest at the end of year one.
−100.00	We remove $100 at the end of the year. This is an amount that we've specified (location C4) that we expect to receive at that time.
$249.53	At the beginning of the second year we start with only $249.53.

+37.43	Our interest at 15% for the money for the second year.
$286.96	Our principal plus interest at the end of the second year.
−200.00	Here we remove $200, as we've specified at location D4.
$86.96	We enter the third year with this amount.
+13.04	Our interest is $86.96 for the third year at 15%.
$100.00	Our total at the end of the third year.
−100.00	As specified at E4, we remove another $100.
$0.00	We're at zero.

These calculations indicate that the net present value of $100 at the end of year one, plus $200 at the end of year two, plus $100 at the end of year three, all at an interest rate of 15%, is indeed $303.94.

Initial investments with cash outflows can be obtained by adding the initial investment to the value obtained from the @NPV function.

@PI VALUE OF π

This function, which has no argument, provides a value of π equal to 3.1415926536. It is used directly in formulas requiring this value; for example, in finding the circumference of a circle whose radius is at location B7, the formula would be

2*@PI*B7

@SIN(ARGUMENT) THE SINE

The sine of an angle, for example, angle *A* of Figure 6-1, is the ratio of the length of side *a* to the hypotenuse, side *c*, that is, SIN

$A = a/c$. The argument is provided in radians, and the sine value will have the appropriate sign (positive or negative), depending on the quadrant in which the angle falls.

Refer to the @ACOS section of this chapter for a brief discussion of radians.

@SQRT(ARGUMENT) THE SQUARE ROOT

This function returns the square root of the argument for an argument equal to or greater than zero, and a value of ERROR for negative arguments.

Refer to the @ERROR section of this chapter.

@SUM(ARGUMENT1,ARGUMENT2,...) SUM THE VALUES

The @SUM built-in function returns one value, the sum of all entries referenced in the arguments. Let's look at a simple example in Figure 6-13 where two columns of numbers are totaled with slightly different formulas at A9 and B9 as follow.

Figure 6-13. Two examples of the @SUM function.

Location	Formula	Explanation
A9	@SUM(A2...A7)	This is a straightforward example of the use of the @SUM function.
B9	@SUM(B1...B8)	In this function the series begins and ends one row beyond the actual values that we want to sum. Here one of those entries is blank (location B1), and the other has a label that evaluates to zero (the ---- at location B8). Therefore, neither affects the result. The advantage of this usage occurs if we want to add or delete a full row. With the formula at B9 we can add or delete rows without the need to revise the formula. We can do the same at A9 unless we want to add or delete the first or last entry of the series. When we want that to occur we must revise the series reference of the function at A9.

Refer to the "Introduction" section of this chapter, where a number of examples of arguments are included. In addition, in Chapter 3, Getting Started, the @SUM function is described in the section "Using one of the Built-in Functions." Chapter 8, Recognizing, Preventing, and Correcting Errors, contains a discussion of some possible problems with this function.

@TAN(argument) THE TANGENT

The tangent of an angle, for example, angle A of Figure 6-1, is the ratio of the length of side a to side b, that is, TAN $A = a/b$. The argument is provided in radians, and the tangent returned will have the appropriate sign (positive or negative), depending on the quadrant in which the angle falls. Angles for which the tangent is not defined, for example $\pi/2$, return the value ERROR.

Refer to the @ACOS section of this chapter for a brief discussion of radians.

Chapter 7

Other Topics

INTRODUCTION

This chapter contains information on a number of additional features or concepts to be considered when using a VisiCalc system. These topics are

- Backup
- Memory
- Accessing electronic sheets from other software
- Circular references
- Forward references

We'll discuss each of these topics and illustrate their importance to using VisiCalc successfully.

BACKUP

Backup is the concept of providing an alternate course of action if some problem appears in the original. For our purposes, we need to consider a relatively wide, systems-oriented variety of problems. Any one of them can halt our successful operation of VisiCalc. If we use a VisiCalc system in areas where its use is crucial, then we need to provide for alternates when problems occur.

Let's look at the components of our system, the potential problems that can develop with them, and methods of backup for each.

System Component	Potential Problem	Backup Procedure
VisiCalc	Our copy of VisiCalc may not work properly, may be misplaced or otherwise lost, or may have been damaged, for example, through mishandling of the diskette.	Study the warranty accompanying the VisiCalc diskette, and if appropriate obtain a second, backup copy of VisiCalc through the vendor or its distributor.
Hardware	Part or all of our hardware may not work properly.	Complete appropriate service agreements and potentially arrange with another user of similar hardware to support each other in an emergency.
Electronic sheets	The diskettes or cassettes on which we have saved electronic sheets may not function appropriately, may be lost, or may have been inadvertently destroyed.	Use VisiCalc to save sheets to more than one diskette or cassette, or use software of the computer system to copy one complete diskette (or cassette) from another. These procedures will provide multiple backup copies of each sheet. Then store the backup copies separately from the original copy. If and when the backup copies are needed, copy them before use so that at least two copies always exist.

Documentation	The printed information we prepare to accompany our spreadsheets may be lost or destroyed.	Prepare photocopies and store them separately from the originals.
Users	If our system depends upon a trained user to enter data or otherwise use the sheet, that person may not be available when needed for a variety of reasons.	Train backup personnel who will be able to use the full system should it become necessary.

What's important is to be prepared for a wide variety of problems that may cause a disruption in the use of our VisiCalc system.

MEMORY

The VisiCalc programs and the single electronic sheet on which we are working are both in memory of the hardware at the same time. VisiCalc itself has a fixed size, therefore leaving us with a fixed area that becomes the maximum size of a sheet that we can enter. As we write on the sheet, the information is stored in memory, reducing the amount available for us to continue writing.

The memory indicator on the right end of the prompt line provides information for us on the amount of memory remaining for our use. In Figure 7-1 notice that the indicator shows 10. On this system, which began with 48 of these units of memory, called kilobytes, VisiCalc itself required 29, leaving a balance of 19, which is the maximum that we can use for each sheet. This particular spreadsheet therefore required 9 of these, leaving 10.

As we continue to build on this sheet, we may exhaust the remaining memory. If this occurs, the memory indicator will change to an "M" and no additional entries will be accepted.

We can control the amount of memory in several ways. We have a choice of the amount of memory purchased for the hardware.

VisiCalc itself is fixed in size. We can't load only the part that we may wish to use; we must always load all of it.

However, we can control in some ways the required memory of the sheets that we build. For example, if we build sheets beginning at the top left of the sheet, we'll use less memory than if we begin in the middle of the sheet.

The major action that we can take when we need additional memory is to take steps to make the sheet smaller. This can include removing blank lines or underlines and eliminating some labels. Documentation on the sheet can be placed on a separate sheet. It may be possible to reorganize the sheet completely so that the problem is solved in smaller parts.

After removing any "extra" data as suggested, it may be necessary to store and then reload the sheet to notice the effect of reducing the memory used. Before reloading, memory should be cleared to prevent overlapping of the smaller and larger sheets.

It's crucial to recognize the difference between VisiCalc sheets in memory and these sheets on an external storage device (diskette or cassette). When we have a sheet on a diskette, we must load it into memory to use it in any way. If we change it while it's in memory, we have not altered the data on the diskette. The copy on the diskette is not changed until we take a deliberate action with the Storage command.

This has both advantages and disadvantages. If we load a sheet and inadvertently destroy part or all of it, the original is still safe on the diskette. At the same time, if we load a sheet, and work to refine it in some way, and then forget to store it, the refinements are lost. We must understand that external storage and memory are different.

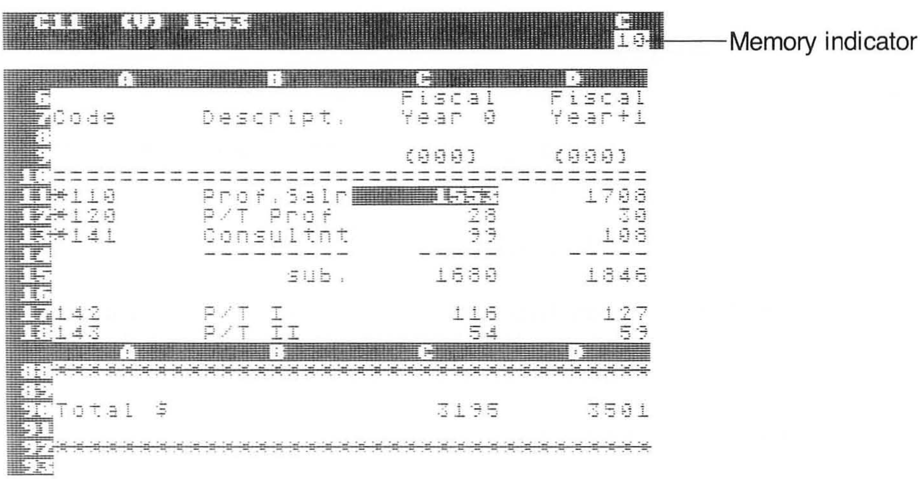


Figure 7-1. The memory indicator shows 10.

Diskettes have the capability to store more than one sheet, which can be a timesaver for us in terms of exchanging, or swapping, diskettes as needed. It also means that backup becomes more important since loss of one diskette may mean that many spreadsheets are lost.

ACCESSING ELECTRONIC SHEETS FROM OTHER SOFTWARE

It is possible for us to prepare software that accesses electronic sheets we prepare and store with VisiCalc. "The Programmer's Guide to the Data Interchange Format" (refer to the Bibliography) contains detailed information concerning the Data Interchange Format, DIF (a trademark of Software Arts, Inc.). This short booklet contains sample BASIC programs that create and list a DIF file.

CIRCULAR REFERENCES

With VisiCalc it is possible to write an entry that refers to itself. For example, at location B3 it is possible to write the formula

(B2+B3)

This is referred to as a circular reference in VisiCalc. Let's demonstrate.

We type	Means
/C	Begin the action to clear the sheet.
Y	Hesitate to be sure that we want to clear the sheet, then type Y.
>B3 RETURN	GO TO B3.
(B2+B3) RETURN	Place this formula at location B3. After entering RETURN, the number 0 appears as the value at this location on the sheet.
>B2 RETURN	GO TO B2.
5 RETURN	Enter a value of 5 for B2. At this point, both B2 and B3 have the value of 5.

! Recalculate. B2 continues to show 5, but B3 shows 10.

! Recalculate. B2 remains 5; B3 becomes 15.

Repeated recalculation results in increasing the value at location B3 by 5 for each ! entered.

Usually this is undesirable and can be avoided by using care and not entering a formula at a location that refers to the location.

On occasion, this capability may be of value. For example, let's begin again with a clear sheet and develop a problem that makes use of a circular reference. We'll begin with the sheet of Figure 7-2, in which the global column width has been set at 12 and the dollar and cents format has been set globally. Column A contains values 100.00, etc., entered from A9 through A13. Column B has had the formula

(A9*B3)

replicated from location B9 through B13 with A9 replicated relatively and B3 with no change.

We're computing interest at a rate of B3 (not yet entered) on principal in positions A9 through A13.

Now let's move to B2, enter .01, then move to B3, and enter .08. We now have the screen in Figure 7-3.

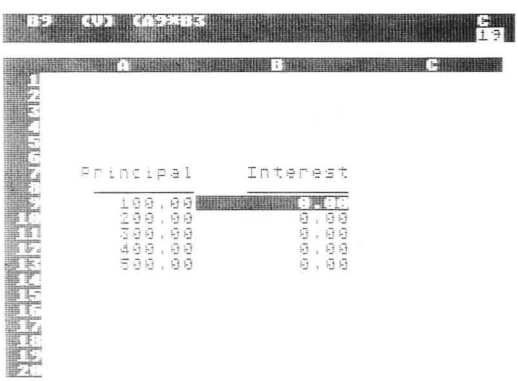


Figure 7-2. A sheet on which we'll demonstrate a possible use for a circular reference.

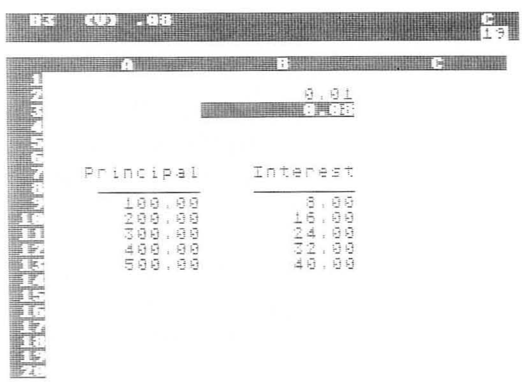


Figure 7-3. A starting value of .08 is placed at location B3.

Now, while still at location B3, let's enter

(B2+B3)

Our value at B3 becomes 0.10. B2 (value .01) was added to B3 (value .08), which changed the value at B3 to 0.9, which caused a recalculation resulting in another increment of 0.01 to B3. The results are in the screen of Figure 7-4. Repeated recalculation, by pressing the exclamation point repeatedly, will cause the regular incrementing of location B3 by .01, and the resulting recomputation of all the INTEREST values dependent on it. After pressing ! several times, the screen in Figure 7-5 was obtained for an interest rate of 14%.

If the sheet in Figure 7-5 is stored and then reloaded, the value ERROR appears at location B3 and subsequent entries dependent upon it. As the sheet is loaded, a value for B3 must be calculated, based on a value at B3 (itself), which does not yet exist. Thus, an ERROR results at B3 and the other entries as shown in Figure 7-6.

In these examples, we see the hazards as well as the potential for using circular references productively. We see the need for caution since inadvertent circular references can easily lead to erroneous results. We have also seen how we can use the technique to generate a series of sheets, each of which may be printed as a table, then recalculated to obtain new values, printed, recalculated, etc. We can similarly use a circular reference to determine quickly the results of changes or to zero in on a value that we're searching for without using a large portion of the sheet.

FORWARD REFERENCES

The topic "forward references" was introduced in Chapter 4, Commands, under the Order of Recalculation section of the Global command. Recall that a forward reference occurs when a formula refers to a value that has not yet been computed. For example, consider entry F8 in Figure 7-7. Intersecting lines have divided the sheet into four sections—I, II, III, and IV—in the figure. If a formula at F8 refers to any location in section I, then F8 should be computed correctly, independent of the global order of recalculation.

If an entry at F8 refers to an entry in section II, then F8 will be computed correctly if the order of recalculation is rowwise. Similarly, a reference to an entry in section III will be computed correctly for a columnwise order of recalculation. In both cases, with the order of recalculation reversed, the entry at F8 may be incorrect unless a recalculation (!) is performed. Finally, a reference

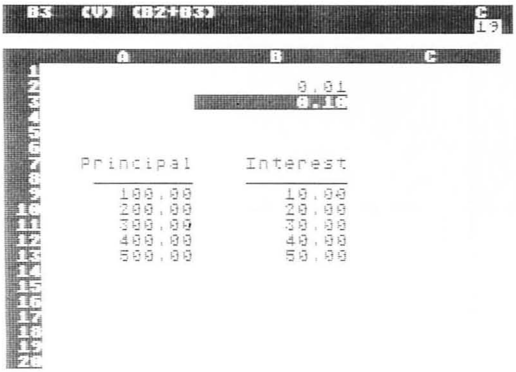


Figure 7-4. The formula at B3 refers to itself and is therefore a circular reference.

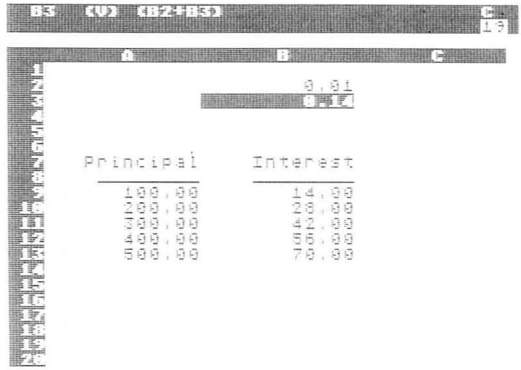


Figure 7-5. At each recalculation (!), the full sheet is recomputed.

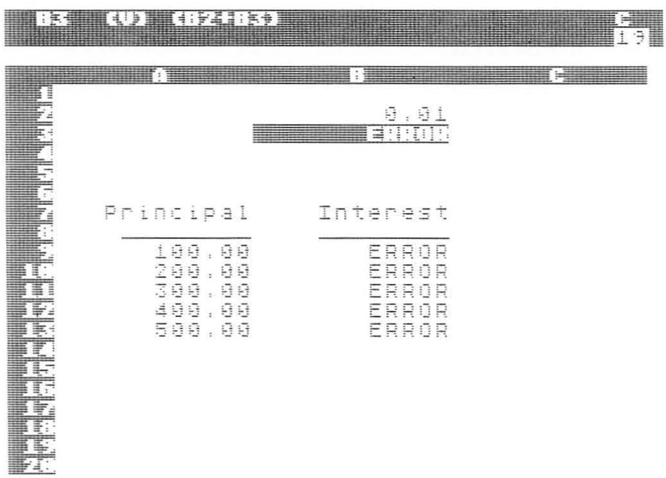


Figure 7-6. When the sheet with a circular reference is stored and reloaded, it contains ERROR values.

to a value in section IV is a forward reference independent of the order of recalculation and may result in an erroneous value at F8 unless a recalculation is performed.

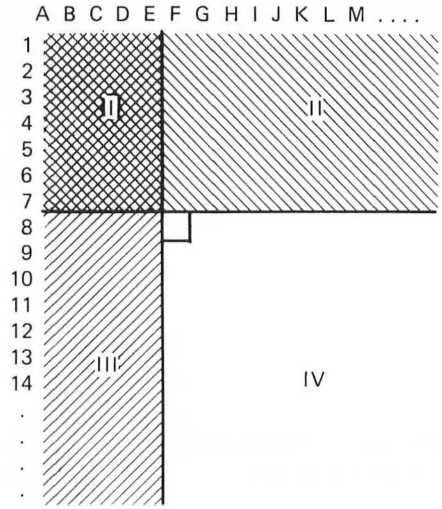


Figure 7-7. We'll study what occurs if a formula at location F8 references entries in each of the four areas marked.

Chapter 8

Recognizing, Preventing, and Correcting Errors

INTRODUCTION

With VisiCalc, as with other computer software, we must continually be certain that the results we produce are correct. We must ask ourselves if our spreadsheet is working properly and if it will continue to work properly. We must work with an attitude that mistakes are common in working with computer systems, and we must accept responsibility for ensuring that steps are taken to reduce the possibility of errors.

To do so we must understand how errors are made with a VisiCalc system. The system from this perspective involves the computer hardware, VisiCalc, the spreadsheets we prepare, the data we provide, the way the sheet is used (by ourselves or others we train), and the final results produced.

We will discuss the types of errors commonly made with VisiCalc, and then the process of locating and changing errors which are discovered through our use of this computer system.

Our approach will be to build spreadsheets that are designed to prevent errors and are self-testing to assure correctness. If we are to depend on the results, we must be certain they are accurate. It doesn't matter if we are working on a one-time problem or on template preparation (Chapter 9) of a sheet that will be used repeatedly.

VISICALC FEATURES DESIGNED TO PREVENT ERRORS

VisiCalc itself contains a number of devices designed to prevent us from making errors. Several common ones are the prompts appearing on the screen and the beep. As an example of the first, if we enter a slash to initiate a command, VisiCalc produces

COMMAND: BCDFGIMPRSTVW-

on the prompt line. This prompt line is designed to help us by providing, after the word COMMAND, the valid character choices available. At this point if we press almost any key other than those above, we will be beeped, indicating that we are being prevented by VisiCalc from making an error.

A third built-in error prevention device of VisiCalc is the requirement of commands such as the Clear command that we confirm our action, here our desire to clear the sheet, by entering the Y key. Similar actions are required at other times by various commands or functions of VisiCalc.

We need to be particularly alert to each signal that VisiCalc issues, whether it is a beep, a prompt, or a request to confirm an action. These are designed to help us. If we're beeped, it is often to prevent an error on our part.

If and when unusual things occur, we need to understand them so that we learn from them, and can use them it to our advantage.

In the remainder of this chapter we will look at some common problems possible with VisiCalc, discuss methods of preventing them, and finally present ideas for how to track and correct errors.

COMMON ERRORS WITH VISICALC

Look carefully at Figure 8-1, in which we are trying to produce a sum (TOT) at the bottom of each column. The sheet contains a number of errors, in fact, only one of the totals appears to be correct. Let's work through each of the columns and explain the cause of the error.

Column B: The formula at location B8 contains

(B5+B5+D5)

It is simply the wrong formula for computing the total. With VisiCalc the ease with which we can enter formulas can also cause problems since it is easy to include the wrong reference.

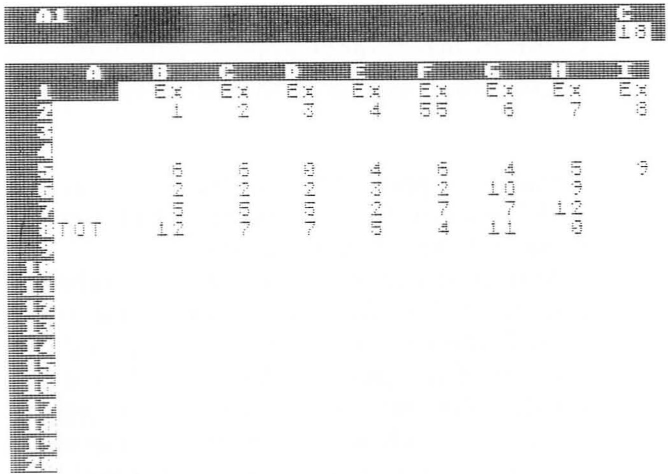


Figure 8-1. A number of examples of errors that can occur with VisiCalc.

Column C: Again the formula is incorrect; here it was entered as

(C6+C7)

In this example, it is possible that at some time during the development of this sheet we were only totaling what is now row 6 and row 7. Then we may have decided to add what shows as row 5. We inserted the new row (row 5) but never revised the formula at C8, causing the incorrect result.

Column D: Finally, a correct result. However, if we looked at the formula at D8, we would see

(D6+D7)

This is "correct" only because entry D5 happens to contain a value of 0. For this reason we do not even know we have an error waiting for us when (and if) we change the value at D5.

This example illustrates clearly the need to be certain that our sheets are functioning correctly. Hopefully, a test of this column with other values would reveal the error. This is particularly important if we depend on this sheet to produce results for a wide range of values over a long period of time.

Column E: Entry E8 contains

@SUM(E6...E7)

and may have occurred for the same reason as the error of column C. An incorrect range appears within the built-in SUM function.

Column F: A look at the contents of location F8 reveals

4

Strange. Suppose that we had carefully placed in that location a SUM function that we are sure had a range of F5...F7, but that is gone and the value 4 appears.

Retracing our steps, we can remember going down the column entering values followed by the arrow keys. We entered 6, then 2, then 7, then 4. At this point we realized the 4 should have gone at the top of column G in location G5. We then moved the cursor to position G5 and correctly entered values down that column.

However, when we entered the value at F8 (incorrectly) we destroyed the formula and replaced it with the value 4.

We see that we must use extreme care in entering values with VisiCalc to ensure that we avoid destroying the formula relationships as we have just done. Because there is no way of "locking" an entry to prevent this, we will need other devices to reduce the possibility of this occurring. Chapter 9, Creating Templates, suggests ways to do this.

Also notice entry F2, which shows 55, a typing error, for it should have shown only a single 5.

Column G: This is a tricky example since the formula at location G8 correctly shows

@SUM(G5...G7)

The error is the data at G6. It looks like the number 10, but in fact it is the digit 1 followed by the letter O not the digit 0. Although contrived for this example, it can happen with VisiCalc because no direct editing capability exists.

Column H: Again we have the correct formula at location H8. It contains

@SUM(H5...H7)

Here the recalculation mode was set at manual, and we simply have forgotten to enter !, which when depressed will cause a recalculation with the resultant correct sum placed at location H8.

Column I: This unwanted 9 was incorrectly left when we thought that we were moving vertically from location H5 to H6 but instead moved horizontally. We realized the error, moved to H6, correctly placed a 9, but forgot to blank location I5.

These examples demonstrate clearly the need for care in the use of a VisiCalc computer system. Our conclusion is to use the power of VisiCalc, while remaining aware of its limitations.

In a number of these examples we solved the problem incorrectly. This was clear in columns B, C, D, and E, where the wrong formula was used. In general, this is an error on our part rather than a hardware, VisiCalc-related, or data problem. We must understand clearly what we're doing and then convert our solution correctly into a spreadsheet.

Additional errors may have occurred because of "forgetfulness" on our part. For example, in the errors of column C and column E we may have forgotten to revise the formula when a row was added to the spreadsheet. In column E the range of the SUM function may be wrong because we read the column location incorrectly from the row and column labels, or we may have moved the cursor to the wrong location.

Column F reveals a completely different type of error, one that is both human and software related. Here we destroyed a formula by incorrectly typing a value over it. This demonstrates a software design problem. We have no way of locking an entry so that it cannot be inadvertently destroyed. This limitation of VisiCalc forces us to use extreme care with every spreadsheet.

ADDITIONAL EXAMPLES OF COMMON ERRORS

Let's look at additional examples of errors common with VisiCalc systems. Again each will require a thorough understanding of the limitations of VisiCalc, of our spreadsheets, of the hardware, of the human users of the system, and of the data.

First, if we enter the number 12345678901234 in a location on the spreadsheet, it is stored as 12345678901200. Those who work with computers regularly accept this, with the acceptance based usually on an understanding of the hardware or software limitations of the system in use. What is significant (in addition to the number of digits stored here), is the fact that the wrong value is stored. Thus, the number of digits stored can clearly affect results. Chapter 5, Labels, Numbers, and Formulas discusses this problem.

Second, if you enter an expression such as

12+6/2

we might expect a result of 15 obtained by dividing 6 by 2, giving 3, then adding that to 12 to result in 15; however, VisiCalc produces a result of 9. This was obtained by first adding 12+6 (giving 18), then dividing the sum by 2 for an answer of 9. Again, those who work with computers regularly know that they must learn the order in which expressions such as these will be evaluated. Different computer languages and software systems handle this differently. VisiCalc users must learn that expressions are evaluated from left to right. This is a simple rule, but it must be learned and understood. A careful user may wish to use, and even overuse, parentheses to indicate clearly what is to occur. Thus,

(12+6)/2

will evaluate to 9 as in this first example. Here the parentheses are redundant but important since they may help us to indicate clearly what we want. If we enter

12+(6/2)

or even

(12)+(6/2)

we obtain 15. We should not hesitate to use parentheses to identify the desired order of computation.

Let's look at another example in this area. If we enter

−(1+5*(4

will it be evaluated, and if so, how? VisiCalc values this as −24. Some other language systems will refuse to evaluate this and might indicate that we have mismatched parentheses or an invalid expression. We should avoid expressions such as the last one, particularly if we're new to programming, as they are a potential source of errors.

Another example is shown next. We may have a column that shows

	1
	1
	1
TOTAL	4

This can occur because of the way in which we use the integer format (/FI or /GFI) and the integer function as discussed for the @INT (integer) function in Chapter 6, Built-In Functions.

These examples show expressions with constant values (6, 12, 2, etc.), but the same results occur if we have entry references (F3, BK100, A3, etc.) in our formulas. In the formula

A3+B4/2

the addition occurs before the division as just explained.

The next few figures illustrate some other common errors that can occur with VisiCalc. In Figure 8-2 look closely at entry B7. These are intended to be dollar ($) values, but we have forgotten to use the appropriate format here, resulting in a value displayed as 18.6 instead of the desired 18.60.

Also look at entry E6. It shows a value of 0 when it should show a series of hyphens. The problem was our use of the Replicate command. We established a formula in entry E4, which was the sum of the entries on the row. When replicated down Column E, the sum was included for row 6, incorrectly placing the value 0 at E6.

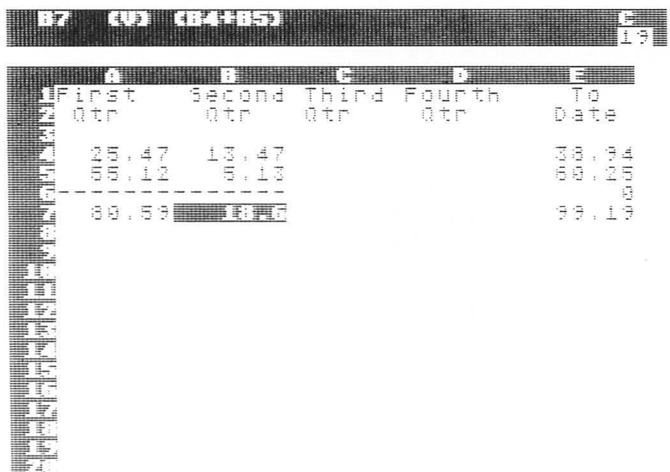


Figure 8-2. A window showing two common errors with VisiCalc usage.

Another common problem occurs with formatting of row and column headings. In Figure 8-3 and Figure 8-4 we have two reports; one printed after the sheet of Figure 8-2 has had the column width reduced and the other after widening the columns. Notice how the headings over the rows are now distorted and misplaced. Also notice in Figure 8-3 that the unformatted entries are displayed with inconsistent, misaligned decimal places.

The lesson of these examples is to postpone some labeling activity until the last steps of spreadsheet preparation. Doing so allows us to prepare correct column and row identification the first

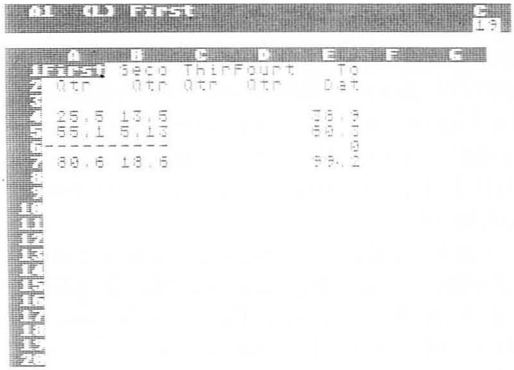


Figure 8-3. Our report after reducing the column width (resulting in incorrect, incomprehensible column titles). The report also displays inconsistent decimal places because of erroneous formatting.

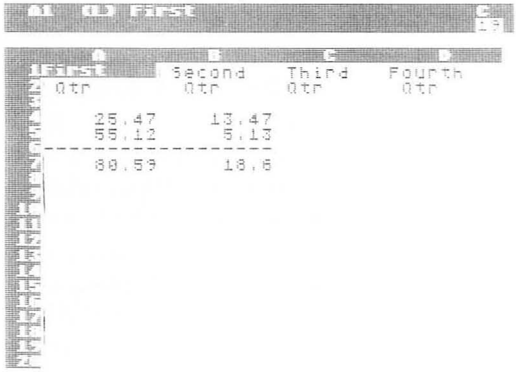


Figure 8-4. The same results as Figure 8-3, but here the column width has been increased.

time. We should determine our formats early in the process so that they can be replicated as needed.

In Figure 8-5 we have printed a report on which the bottom has been omitted by improperly specifying the lower right corner of the report during a Print command. Also notice that the word FOURTH has been improperly spelled (FORTH).

First Qtr	Second Qtr	Third Qtr	Forth Qtr	To Date
25.47	13.47			38.94
55.12	5.13			60.25
------	------	------	------	------

Figure 8-5. A printed report in which the bottom has been omitted by improperly specifying the lower right of the sheet in the Print command. We also have a misspelling (FORTH).

Let's briefly mention additional problems that can occur with VisiCalc.

• We can misuse the commands. The Replicate command, probably the most complicated and powerful of VisiCalc, can cause a number of problems that we discussed in Chapter 4, Commands. Misuse of the N and R prompts of that command can easily cause problems, as can incorrectly specifying source and target locations.

- Because we cannot see the full sheet on the screen, there may be unusual erroneous entries that we are not aware of and cannot easily detect.

- It's easy to point to the wrong entry in completing a formula. If we replicate from this initial incorrect value, we are replicating an error.

- If we unintentionally load one sheet onto another without clearing the screen, we will end up with a spreadsheet that is usually an erroneous combination of the two.

- Circular or forward references (refer to Chapter 7, Other Topics) may cause undesired results as may an incorrect order of computation (refer to the /G section of Chapter 4, Commands).

Throughout this chapter we have illustrated problems that can cause errors for us. Now let's move to a discussion of detecting, correcting, and, even more important, preventing errors.

DETECTING AND PREVENTING ERRORS

Our goal should be to produce accurate information. Although we want to take advantage of the speed of spreadsheet preparation, we do not want to do so at the cost of accuracy. What we have seen is the extreme need for caution. For example, it's very easy to destroy a formula by writing a value over it and not even realize that this has occurred.

In writing programs or in using software other than VisiCalc, a major portion of the program may be devoted to checking input data to make sure that they are valid. For example, if we are preparing a payroll report, we want to be certain that the input field containing the number of hours the employee worked, in fact contains a number. If it does not, we probably have an error. If it has a number, we may want to check on its reasonableness in some way. For example, if the number is generated from a weekly employee, we may not want to process it if it's greater than 70 hours without independent verification.

Even though we will suggest a solution for this particular problem later in the chapter, we must recognize that this editing and checking process is not conveniently available with VisiCalc. We must be especially alert during the data entry activity with the sheet; that is, the user provides this validation function, not the spreadsheet.

In Chapter 9, Creating Templates, we'll develop a series of guidelines to help us include steps that can be taken with VisiCalc to avoid these problems. In the remainder of this chapter we'll look at some examples of actions that we can take to prevent and detect errors that could occur with our sheets.

The QUARTERLY REPORT of Figure 8-6 is a common type of report for which VisiCalc can be used. We show four columns of information with subtotals (SUB) included for each and a grand total (TOTAL) at the bottom of each column. Notice that we've filled the sheet with numbers for which it is easy to determine if the sheet is working correctly. This is certainly not an exhaustive test, for we have seen earlier in the chapter that a large number of things could go wrong.

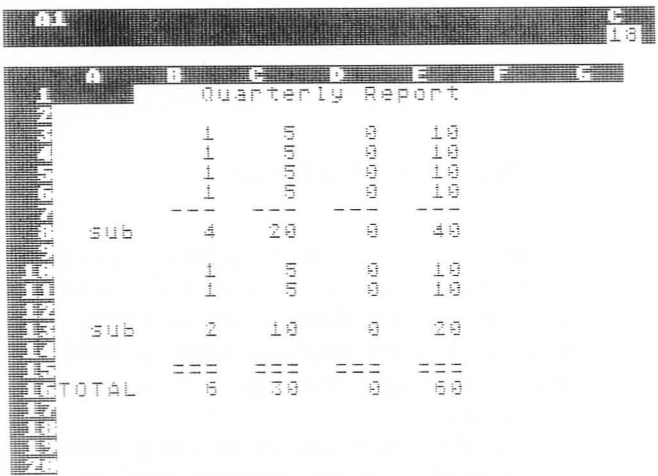
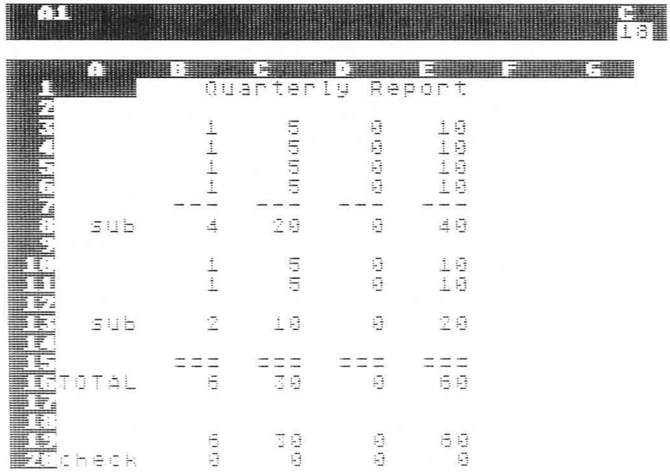


Figure 8-6. A typical VisiCalc sheet with test data but no double-checking of the totals.

Let's demonstrate a simple way of providing one check on the accuracy of our work. In Figure 8-6 the TOTAL was computed by adding together the SUB totals of rows 8 and 13. Thus, the formula at entry B16 contains

(B8+B13)

As a double check, look at Figure 8-7. Two rows have been added to the report, rows 19 and 20. The formula at entry B19 is



	A	B	C	D	E	F	G
1		Quarterly	Report				
2							
3		1	5	0	10		
4		1	5	0	10		
5		1	5	0	10		
6		1	5	0	10		
7		---	---	---	---		
8	sub	4	20	0	40		
9							
10		1	5	0	10		
11		1	5	0	10		
12							
13	sub	2	10	0	20		
14							
15		===	===	===	===		
16	TOTAL	6	30	0	60		
17							
18							
19		6	30	0	60		
20	check	0	0	0	0		

Figure 8-7. The sheet of Figure 8-6 after revision to perform a double check on the totals.

(B3+B4+B5+B6+B10+B11)

We have separately added all the entries used to compute the SUB totals. Finally row 20 contains the difference between row 16 and row 19; that is, entry B20 contains

(B16-B19)

As we manipulate the values in this sheet, we will watch row 20 carefully to be certain that it contains zero values across the row. Another simple way to do this is to place the formula

@SUM(B3...B15)/2

at location B19. All non-numeric values are ignored (since they have value 0), and since each value is included twice, division by 2 gives a sum of the column.

If we print a report from this sheet, these last two rows can be omitted if desired.

We have built a simple check into our sheet. It is redundant and can be time-consuming on a large sheet, but techniques like this are extremely important in order to detect and prevent errors.

Let's look at another technique that can be helpful as a testing device. We can create temporary rows or columns that can help us determine if information is entered correctly.

For example, if we have one column for each of the 40 people in one department, we can quickly insert a row above the names and place the values 1, 2, 3, ... 39, 40 across this new row. If 40 appears over the last name, we know that we have 40 columns (although we may have a duplicate among them). If 40 does not fall over the last name, then this testing device has been helpful in identifying an error that we can find and correct. The new row can be deleted, if desired, after serving this testing function.

For a second example, let's return to our payroll problem. In Figure 8-8 we've shown part of a spreadsheet including employee names, the number of hours each worked this week, and a column labeled CHECK (column C). The formula at the highlighted entry, C4, contains

@MAX(B4−70,0)

A similar relative formula exists at C5 and C6. Let's follow the action for the entries shown. Remember our purpose: we want to know any employees for whom our data shows more than 70 hours worked. On the screen, MARIAN shows 80 hours in entry B4, that is, above the exception level of 70. For her the formula evaluates as

@MAX(B4−70,0)
or @MAX(80−70,0)
or @MAX(10,0)
or 10

For NOAH, who worked 40 hours, it evaluates as

@MAX(B5−70,0)
or @MAX(40−70,0)
or @MAX(−30,0)
or 0

GABRIEL is evaluated similarly, to value zero (0).

Thus, we have created, in the C column, an exception report, that can be scanned for nonzero entries, each of which indicates an individual who worked for over 70 hours this week.

It is important to recognize that although we have automated this calculation, the exception recognition shown in column C is still dependent upon an individual's scanning that column for nonzero entries. Thus, the process of identifying possible errors is still heavily dependent upon the user of the system. Although this is ultimately true of most computer systems, other systems provide the capability for exception reporting, etc., which is not available with VisiCalc.

In this chapter we've seen several concepts in testing, including checking with simple numbers, and built-in checking. The latter

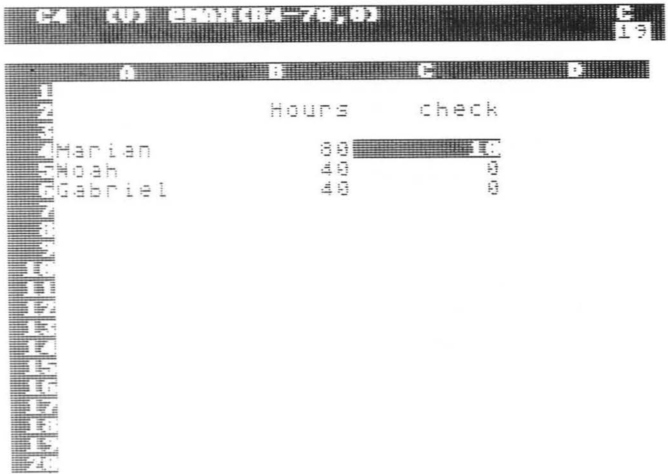


Figure 8-8. An example in which a CHECK column has been added to check on the validity of the input data.

indicates the important concept of considering testing when the sheet is constructed.

Let's list a number of other concepts that can help produce correct results.

- Create templates using the concepts in Chapter 9, Creating Templates, and use these techniques even for one-time use. That chapter contains suggestions for avoiding many of the problems discussed in this chapter.
- If the same spreadsheet is used repeatedly for different test values, consider clearing and then reloading the sheet for each new group of values. If we have inadvertently destroyed part of the sheet when entering one set of data, this action will limit the damage to that one set rather than that set plus all that follow.
- In addition to testing the sheet with these simple values, also test with "large" numbers to ensure that column widths for all entries are wide enough to display expected values in an appropriate format.
- Test our results against results that we have computed by hand or with other programs that can serve as controlled comparisons. Do this for test data and for real data alike.
- Follow the cursor movement carefully and be certain that we are at the correct location before entering a value.

- Use the Replicate command to help fill entries with test data.
- Have someone else test our system. This might include working with the person while he or she begins with our documentation (refer to Chapter 10, Documentation) and works through to the final steps of report preparation. We'll need to be alert to parts of the system that cause problems and then revise them to produce a better system. If others will regularly use our work, this training step is very important for both ourselves and the user (refer to Chapter 11, What Our Client, Secretary, or Supervisor Needs to Know).
- Recognize the limits of VisiCalc. As we have seen, it may not be appropriate for all problems that we wish to solve with a computer system.

CORRECTING ERRORS

Once our testing reveals an error, we need to locate and correct the problem. We should be delighted that the error has been found during testing and not later, after we acted on what we believed to be accurate information. We will need to be careful that in locating and correcting an error we don't inadvertently create additional errors. When we believe that we've corrected an error, we'll need to begin our testing again.

Debugging, the process of locating and correcting known errors in a spreadsheet, will usually begin with an examination of the entry at the locations known to be incorrect. If these entries appear to be correct, we can examine entries that are referenced in formulas contained at the locations known to be incorrect. This step-by-step process will hopefully reveal the problem. If it is an error detected at a location created by a replication process, then all corresponding entries may need to be replicated again with the correction.

SUMMARY

VisiCalc itself provides limited assistance in the processes discussed in this chapter. We are limited to examining contents of entries one by one. We do not have cross reference listing facilities; that is, we cannot determine all entries that reference a particular entry. We cannot tell which entries of the sheet are blank and which have contents without moving over the entire sheet entry by entry.

This chapter has listed a broad range of possible errors, ranging from incorrect expressions, to destruction of formulas, to solving the wrong problem.

Chapter 9

Creating Templates

INTRODUCTION

One of the most powerful features of VisiCalc is the ability it provides to create templates. Templates are patterns or models that guide a person in entering data onto the electronic sheet. A template is an electronic sheet on which we've written labels, formulas, and possibly some, but not all, of the values needed to evaluate the sheet fully. We'll store this sheet and later, when we need to get information involving this sheet, we'll load it, enter the missing values, and instantly have our information.

Templates also allow for the separation of the data entry function from the steps involved in defining the relationships between the entries. This is extremely important, as we'll see.

Let's look at the simplified expense budget in Figure 9-1. In looking at this report we can't tell if relationships exist between the entries. How are PERSONNEL dollars computed? Are they a function of EMPLOYEES (the number of people working for us)? Are BENEFITS related to PERSONNEL costs, EMPLOYEES, both, or neither? Was TRAVEL computed as $300 per person?

It's clear that the report reveals nothing of the functional relationships between the entries. If we intend to use this spreadsheet for new calculations we can do so only if we are intimately familiar with which entries we can change. We need to know which hold values that we are to enter and which hold formulas that we do not want to change. If we prepared the report three months ago, it's possible that the details may now be hazy. What if we ask an employee to work with the sheet and the individual is not familiar with it? What if our manager wants to use this sheet? What if the sheet has hundreds of entries instead of the few of the example?

Creating a template as the first step of solving a problem can alleviate many of these potential difficulties. Let's follow a number of iterations in the process of creating a template for the example of Figure 9-1. As we do so, we'll develop a series of template guidelines. These guidelines will consist of a mixture of common sense, an understanding of the capabilities and limitations of VisiCalc, and a knowledge of what computer programmers and users have learned (primarily the hard way) about interacting with a computer system. This last category will be extremely important to VisiCalc users who do not have previous experience in writing and regularly using computer systems.

Let's start. We will built templates for several different examples, starting with the problem of Figure 9-1.

```
        Simple Budget

Employees                50


Personnel           1000000
Benefits             250000
Telephone             10000
Rent                  10000
Travel                15000
Hospitality            1000
Equipment             18000

          TOTAL:    1304000
```

Figure 9-1. A simple expense budget example for which we'll create a template.

GUIDELINES FOR CREATING TEMPLATES: EXAMPLE 1

Template Guideline 1: Understand the problem thoroughly.

The ease of using VisiCalc can tempt us to start filling the blank sheet without a complete understanding of the problem to be solved. In general it's a mistake, but not always, as we'll see. The new computer user is probably developing an understanding of the explicit nature of the directions that must be provided to a computer to receive the results desired. This is as true with VisiCalc as with any other method of programming or communicating instructions to a computer.

On the other hand there are often problems that we try to solve but which we know we do not understand well. Tools like VisiCalc can be helpful for this type of problem since they permit, or even encourage, experimentation with different model solutions. For this type of problem VisiCalc can become a working tool to help us develop a thorough understanding of the problem.

What's important is that we be able to distinguish between the problems that we do understand (or should understand) and those we know we don't understand.

Let's apply this guideline to our sample budget problem. Our budgeting assumptions here will be simple and straightforward, and we'll assume that we thoroughly understand our problem. Each of the fields is briefly explained as follows:

Field	Formula, variable, or constant	Notes
EMPLOYEES	variable	We'll enter this field. Figure 9-1 shows 50 people working for us. This is a variable, meaning that we'll enter a numeric value for this field. When we develop our template we'll leave a "place-holder" for it, as we'll see.
PERSONNEL	formula	We compute this as $20,000 times the number of EMPLOYEES. We'll place the formula in our template.
BENEFITS	formula	Compute as 25% of PERSONNEL costs.
TELEPHONE	variable	We'll enter this value.
RENT	constant	Although VisiCalc doesn't distinguish between variables and constants, the distinction can be important in templates. A constant is a value that will be entered by the person who creates the template and not by the person who later uses this template.

		For our problem we know that RENT will be a fixed cost and we'll place the $10,000 directly into the template.
TRAVEL	formula	Compute as $300 per person.
HOSPITALITY	variable	We'll enter this.
EQUIPMENT	variable	We'll enter this.
TOTAL	formula	The sum of the seven expense fields.

Let's create a template for this problem. For each of the four variables to be entered, we'll place a series of periods (.......) in the entry. This field is the "place holder" that we mentioned. We'll do this here with the use of the Repeating Label command by typing

/-.

This fills the field with periods. In a later section of this chapter we'll discuss other ways of identifying such a field as one in which a variable should be entered. The use of repeated periods is only a convenient, visual way of marking the field and is not related to any VisiCalc operational requirement. Let's identify this concept as our second guideline.

> **Template Guideline 2:** Clearly identify all fields for which data is to be provided.

In creating our template, the formulas, labels, and constants (such as RENT here) are entered normally on the sheet. Figure 9-2 shows our template. Let's call it a template-in-process, because we'll be revising it significantly.

Notice that a number of entries in which we have placed formulas show 0. As we know, when VisiCalc encounters a label in the field of a formula the label is treated as a zero value. Here, for example, PERSONNEL has been entered as a formula (20000 multiplied by the number of EMPLOYEES), and the EMPLOYEES field contains .... Therefore, PERSONNEL is evaluated as

20000 * 0

or 0, the value shown in Figure 9-2.

The zero-evaluation of labels by VisiCalc will be very useful in our work on templates.

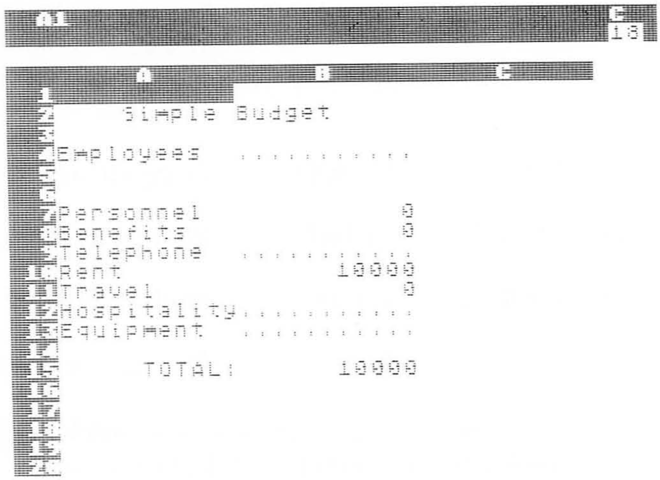


Figure 9-2. Our template-in-process for our budget example.

We are beginning to see the power of a template. The variables that need to be entered are clearly identified (here by periods).

In Figure 9-3 we have entered the four values and have our new budget. We've done this by

- Loading the template.
- Positioning the cursor in turn over the four locations requiring an entry (indicated by a series of periods).
- Entering the values:

EMPLOYEES	=	40
TELEPHONE	=	5000
HOSPITALITY	=	1000
EQUIPMENT	=	10000

After following these few steps, we can view our new budget as a function of these four new values.

Let's continue to develop this same template and introduce some additional guidelines.

Template Guideline 3: Include documentation within the template.

Documentation is a broad term in the computer field encompassing a wide variety of written materials intended to be helpful to a user of all or part of a computer system. It may be program documentation, operating documentation, system documentation, user documentation, etc. Chapter 10, Documentation, discusses the

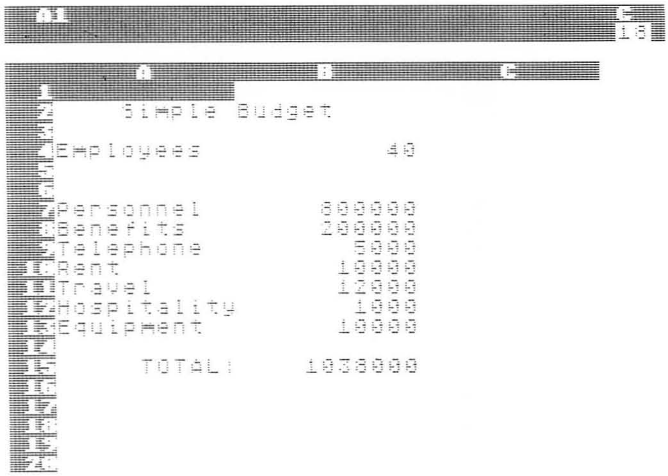


Figure 9-3. A new budget after entering four values on the template that we've created.

topic more fully from a VisiCalc perspective. In our example we'll start by discussing documentation for a person using the templates we'll create.

Our environment and intended use of a template should provide guidance for the appropriate level of documentation. If we use this template several times within a short period and then discard it, we may need little documentation. However, if our template is used by a large number of people over a long period of time, we need to provide more extensive documentation. If we use the template infrequently, then we need to provide enough documentation for ourselves to ensure that we can successfully use or revise this template when we need it next. Figure 9-4 suggests the beginning of modest documentation, which here has been placed in the top left corner of the sheet. Included are the file name under which the template has been stored, here FIG94, and the name of the author of the template, here DON BEIL. As we proceed, we'll add additional documentation to our templates.

Let's use our template again. Assume that we need revised budget information because of potential changes in the number of EMPLOYEES and other expenses. Suppose the projected values are

EMPLOYEES	=	50
TELEPHONE	=	9000
HOSPITALITY	=	1000
EQUIPMENT	=	30000

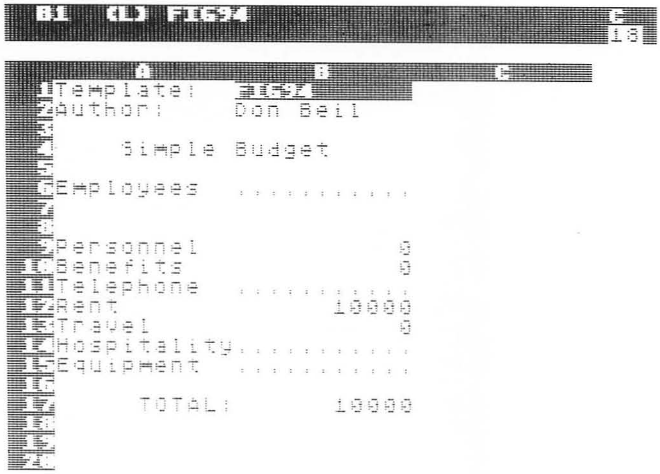


Figure 9-4. Minimal documentation has been included in the first entries of the template.

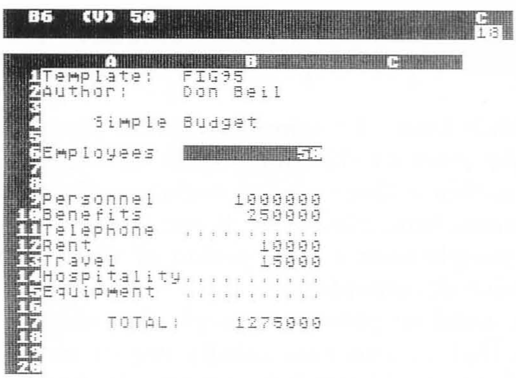


Figure 9-5. Our template after partially entering values. The EMPLOYEES value has been entered.

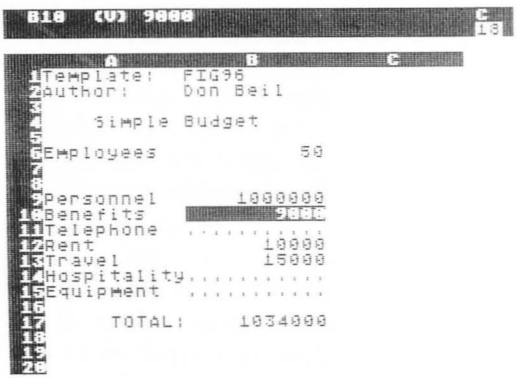


Figure 9-6. Our template after inadvertently destroying the BENEFITS entry by writing our TELEPHONE entry there incorrectly.

Figure 9-5 shows our template after we've partially entered the data. In the figure we've entered the number of EMPLOYEES (note the cursor position) and can see that PERSONNEL, BENEFITS, and TRAVEL all have been computed as a function of EMPLOYEES. Then, in Figure 9-6 we enter the $9000 TELEPHONE expense; however, look carefully at the cursor location. In error the number has been entered in the BENEFITS location, destroying the correct value.

Not only have we destroyed the value at the BENEFITS entry, but we have also destroyed the formula that created the relationship between BENEFITS and PERSONNEL. This error leads us to our next two guidelines.

> **Template Guideline 4:** Back up templates on diskettes (or tapes).

As templates are developed, they should be saved periodically to ensure that work is not destroyed. It's a point that's been made several times and that will be made again later in this book.

The example of Figure 9-6 demonstrates clearly the need to have backup of files. In this case we've destroyed the formula for BENEFITS by placing a value incorrectly in its entry location. If we know the relationship, we can reenter the formula. But if we're entering data on a template that someone else has written, or someone else is using our template, there are problems.

The backup is crucial. If we inadvertently destroy a template, we can reload it from the last backup if it has been saved. In fact, this is one of the values of templates; they can be saved and loaded as necessary to obtain desired results.

This example also leads to the next guideline.

> **Template Guideline 5:** Separate the data entry process from the calculations.

There may be no concept more important than this one in creating a template. The ease with which values, labels, or formulas can be entered onto the sheet is both an advantage and a disadvantage in using VisiCalc.

In Figure 9-6 we destroyed a formula by replacing it with a value in a single potentially devastating move. Such an action can easily go undetected with VisiCalc. It requires extreme care in entering data, a point we'll return to later.

It is important in considering template creation to understand the difference between the processes performed to generate a template and the processes followed to use a template. In discussing data entry, we're discussing using an existing template (not generating a new template). With VisiCalc, data entry is the process of placing values onto predetermined locations of the electronic sheet. As designers it's our responsibility to create templates that make this second process as easy and foolproof as possible.

The last guideline suggests that we can take actions in the design of our template to reduce the likelihood of such a destructive action. Figure 9-7 shows the same template on which we have created a separate area for data entry located below the report. We'll enter values in a separate section of the sheet that functions only as the data entry area. To add this area to the sheet, we must change our formulas.

```
Template:     FIG97
Author:       Don Beil

      Simple Budget

Employees                 0


Personnel                 0    Report area
Benefits                  0
Telephone                 0
Rent                  10000
Travel                    0
Hospitality               0
Equipment                 0

      TOTAL:          10000


Employees    ...........
Telephone    ...........       Data entry area
Hospitality...........
Equipment    ...........
```

Figure 9-7. The budget template in which we've separated the data entry area from the calculation report area.

The changes in formulas have occurred as follows:

Entry	Figure 9-4 entry contained	Figure 9-7 entry contains this formula
B6 (EMPLOYEES)	....	(B22) or +B22
B11 (TELEPHONE)	....	(B23) or +B23
B14 (HOSPITALITY)	....	(B24) or +B24
B15 (EQUIPMENT)	....	(B25) or +B25

We can see that we've replaced our series of periods (..........) with a simple formula reference to the data entry area of the template. These formulas are essentially four copy instructions. Visi-

Calc copies the data entered in the lower portion of the screen to the desired report format area at the top of the screen.

This action alone can reduce the possibility of destruction of the formulas forming the relationships in our budget template, and as such is a valuable addition to the template.

Additional steps can be taken to isolate the data entry function, as we'll discuss.

Template Guideline 6: "Write protect" the report area.

VisiCalc doesn't currently provide a method of "write protecting" or "locking" an entry (or entries) of the sheet. By this we mean that we can't indicate that an entry of the sheet, perhaps a formula or constant, can't be changed without a special step on our part. This would be a desired feature for template creation since it could prevent the accidental destruction of formulas such as we caused in the template of Figure 9-6.

Even so, there are several steps that we can take to give a similar (but not as effective) protection. We can include titles in the data entry portion of the sheet, and fix them both vertically and horizontally. This provides a level of protection against accidentally moving into, and subsequently writing on, the report portion of the template. We'll demonstrate this and combine it with the two following guidelines.

Template Guideline 7: Build redundancy into the template.

Template Guideline 8: Store the templates so that when reloaded the cursor is positioned to the first value entered.

Figures 9-8 and 9-9 summarize the last several guidelines. Notice in Figure 9-8 that two rows have been inserted above the data entry section. These lines, which are redundant, are

TEMPLATE: FIG98
AUTHOR: DON BEIL

They are the same as the top two rows of this template but serve to provide information in the window that the person entering data will see, as we'll discuss.

Figure 9-9 shows the value of this redundancy. It is the image that will appear when an individual issues a Storage command to load the file FIG99. In this way we have built redundancy into our template and have stored the template so that when reloaded, the cursor is positioned at the first value entered. The redundancy is an extra effort but provides additional information and documentation for the user. We have another example of redundancy

```
Template:    FIG98
Author:      Don Beil

     Simple Budget

Employees                  0


Personnel                  0
Benefits                   0
Telephone                  0
Rent                   10000
Travel                     0
Hospitality                0
Equipment                  0

       TOTAL:          10000
```

Desired report

```
Template:    FIG98
Author:      Don Beil
```

New lines (redundancy)

```
Employees    ...........
Telephone    ...........
Hospitality...........
Equipment    ...........
```

Data entry area

Figure 9-8. A printed listing of our template to which we've added two redundant lines of documentation.

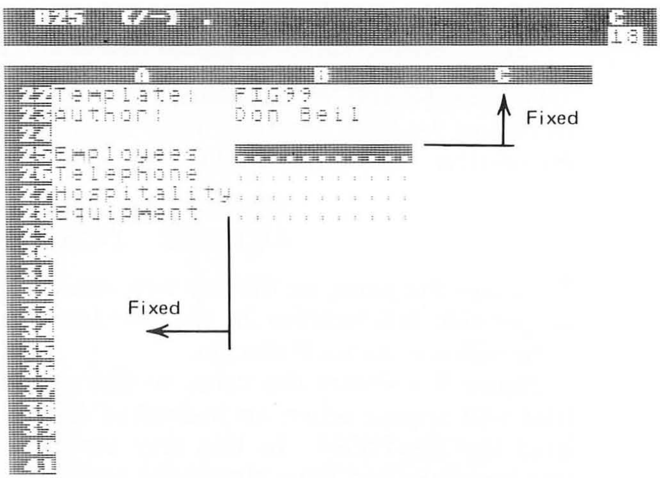


Figure 9-9. The window that appears when loading the budget template. This is the data entry area of the sheet on which the titles indicated have been fixed.

with our four variables entered in one location and copied automatically into another. We'll see other examples later in the chapter.

The titles have been fixed horizontally and vertically as marked in Figure 9-9, and the cursor comes up on the screen as shown, ready to accept the first value to be entered. If the person entering data accidentally attempts to move up the screen or to the left, the user will be beeped and prevented from destroying entries other than those they can reach by "arrow access." That is, with the arrow keys, the user can only access areas we established for data entry. Use of the GO TO command, however, will enable the person to move into other areas of the sheet when necessary (as we'll see). Since it's unlikely that GO TO would be issued accidentally, we've created a limited form of protection for the important upper area of this sheet.

With this technique it's possible to achieve some success in "write protecting" our sheet. Another technique is demonstrated later in the chapter.

The screen of Figure 9-9 also demonstrates another guideline.

> **Template Guideline 9:** For data entry, reveal only as much of the sheet on the screen as necessary for this process.

Compare Figure 9-9 with Figure 9-10. The first is a simple screen containing only the area needed for data entry. Much of the

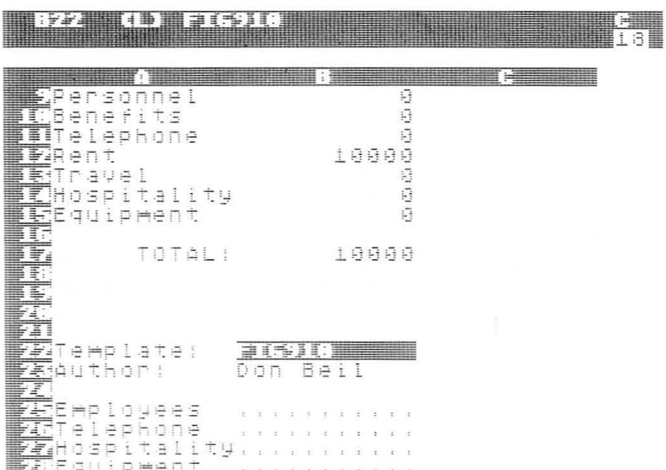


Figure 9-10. An example of a window that may be confusing to a person entering data on the sheet.

screen is blank and easy to understand and to use. Figure 9-10 on the other hand reveals a larger number of entries (almost the full sheet here) and is confusing. For example, a number of the fields occur twice on the same screen. For this reason, the simple screen of Figure 9-9 is preferable.

This last guideline works counter to one of the strengths of VisiCalc, that of observing the instantaneous effect of a change in one field on another entry. Because of that, this particular guideline will be meaningful for some templates but possibly not others.

What's important is to design the data entry function to be simple, direct, and easy to use for the person (including ourselves) entering the data. VisiCalc demands the full attention of the user entering data; therefore templates should be designed not to distract from this process in any way.

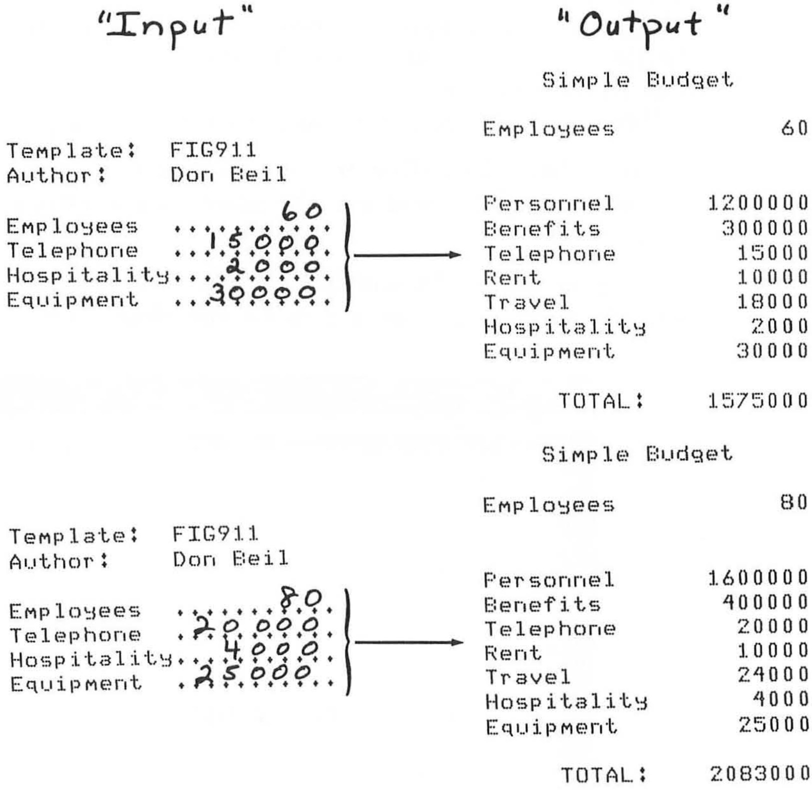


Figure 9-11. A demonstration of recording variables on preprinted "input" forms prepared by printing the data entry segment of the template and then using these forms as a reference for entering data onto the sheet to obtain the reports shown as "output."

Let's think again of the person entering the data, and assume that we have a printer available. Since we've isolated the values to be entered, we could get printed copies of this blank data entry section of the sheet and follow the next guideline.

> **Template Guideline 10:** Use the data entry layout of the sheet for the external data collection.

This is demonstrated in Figure 9-11. Suppose that we want to prepare printed reports for several budget options and that another person will be preparing these reports for us. To communicate our wishes to the person doing the data entry, we can simply write the values we want onto "input" forms printed from the data entry portion of the template as shown in Figure 9-11 on the left. On the right are our finished reports.

We can think of this as using a portion of the template in printed form to prepare "input" for a report generated by VisiCalc.

To assist further in this process, and to build on the report preparation activities of Figure 9-11, let's introduce several more guidelines.

> **Template Guideline 11:** Include instructions in the template.

In Figure 9-12 we've done that. Notice the new last row. It directs the person to GO TO A3 after the data entry function for the four values has been completed. A person entering values vertically down the template will encounter the last row and can

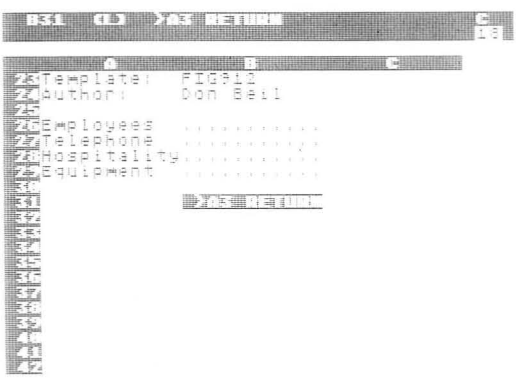


Figure 9-12. Our template with instructions (>A3 RETURN) included.

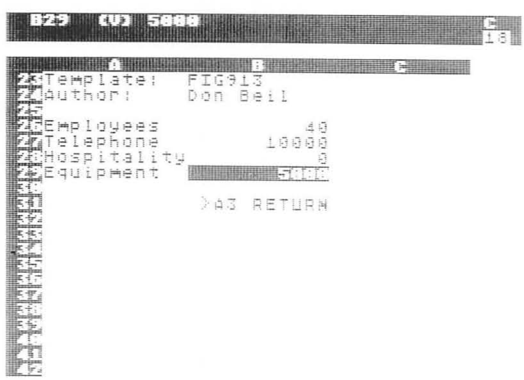


Figure 9-13. Our template after entering the four values and before following the instructions of the last row.

follow the directions included there in the form of a VisiCalc command.

This label was placed at this entry by entering the double quote key (to indicate that a label is being entered), followed by the characters shown in the entry. That is, we entered

">A3 RETURN

In this example, assume we have entered the four required values, as shown in Figure 9-13. We then execute the GO TO A3 as directed. Following this command, the screen appears as in Figure 9-14.

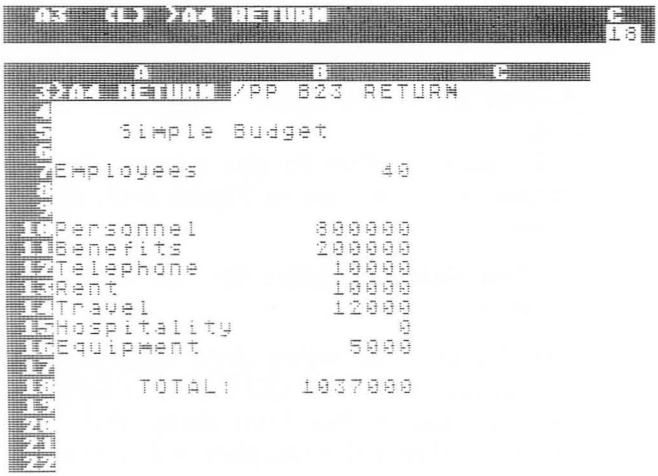


Figure 9-14. Our screen after executing a GO TO A3. We are given additional instructions.

Here a line of additional instructions is included, directing us to move to the next line

>A4 RETURN

and then print the report

/PP B23 RETURN

This is the series of keystrokes to print a report from the location A4 (current cursor location and the desired top left corner of the report) to location B23 (the bottom right corner desired).

When the instructions of row 3 are followed, the report of Figure 9-15 is produced. Notice the last line printed:

TEMPLATE: FIG915

This line serves as excellent documentation for the printed report itself. It should be circulated with the report since it can be a needed reference to the particular template used. Let's produce a guideline from this concept.

```
          Simple Budget

Employees                40


Personnel            800000
Benefits             200000
Telephone             10000
Rent                  10000
Travel                12000
Hospitality               0
Equipment              5000

        TOTAL:      1037000



Template:   FIG915
```

Figure 9-15. The report printed by following the instructions in the top line of Figure 9-14.

> **Template Guideline 12:** Label all printed reports with identification of the template used to produce it.

In addition, it's often desirable to know the "assumptions" from which our report has been prepared. With the layout of the sheet that we've developed, it's easy to add this to any printed reports by simply printing a larger part of the full sheet. This has been done in Figure 9-16 after adding one row with the word ASSUMPTIONS. The variable values used to create this report are clearly indicated at the bottom of the report.

A summary of our final template for this example is shown in Figure 9-17. A number of things have been marked including

- Several areas of documentation.
- Several lines of instructions.
- The format of the first printed report including the template identification (printed report 1).

```
                Simple Budget

        Employees              40


        Personnel          800000
        Benefits           200000
        Telephone           10000
        Rent                10000
        Travel              12000
        Hospitality             0
        Equipment            5000

             TOTAL:       1037000



        Template:   FIG916
        Author:     Don Beil
             "ASSUMPTIONS"

        Employees              40
        Telephone           10000
        Hospitality             0
        Equipment            5000
```

Figure 9-16. A printed report listing our "assumptions" on which the report is prepared.

- The format of a second report, which includes the printed "assumptions" used to create the spreadsheet (printed report 2).
- The data entry window, which appears when the template is first loaded and which in printed form can be used for external data collection.

An additional important concept follows:

Template Guideline 13: Provide documentation for the user other than that written onto the template.

Chapter 10, Documentation, discusses this topic and suggests methods appropriate for VisiCalc users.

GUIDELINES FOR CREATING TEMPLATES: EXAMPLE 2

For the second example, assume that we need a template that will help us at our manufacturing plant to record our production

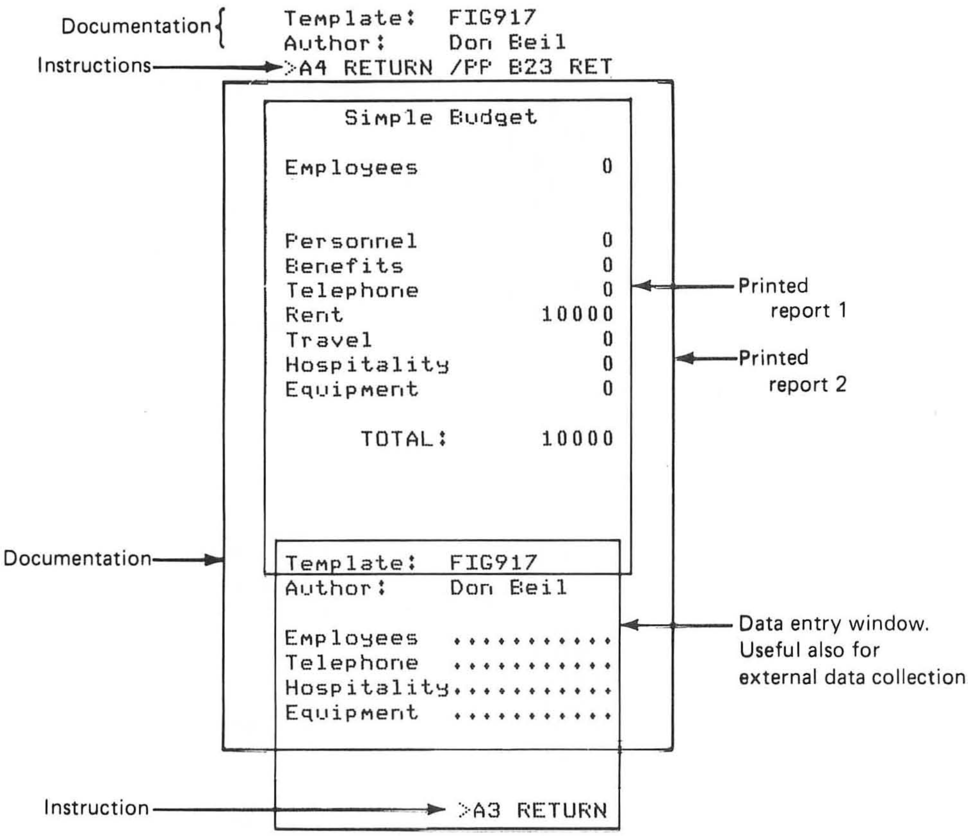


Figure 9-17. A summary of our template.

activity on an hour-by-hour basis for one week and to establish hourly production goals for the coming week. Figure 9-18 shows the desired report that we want to produce.

The values for PRODUCTION THIS WEEK in Figure 9-18 are our actual production figures that we enter late Friday afternoon. Our FORECAST will use a simple goal mechanism. We want to increase production by a specific number of units per hour for every hour. The variables that we'll provide to the template include all the values enclosed in the two boxes of Figure 9-18. All other values will be computed by the template. Notice that we really have two reports on our template. Let's formalize this idea.

Template Guideline 14: More than one report can be prepared on the same template.

If we wish, we can prepare several reports on the same sheet. We can add a simple graph of the same data if desired. On Figure

Production This Week
(000)

	Mon	Tues	Wed	Thur	Fri	TOT
8:00	0	10	10	20	20	60
9:00	2	30	25	20	30	107
10:00	20	30	25	20	30	125
11:00	20	30	40	30	40	160
12:00	18	25	35	30	30	138
1:00	30	30	40	20	30	150
2:00	30	30	40	20	30	150
3:00	30	25	30	30	30	145
4:00	25	25	30	30	30	140
TOTAL	175	235	275	220	270	1175

Forecast:
Production Next Week
(000)

Planned Increase of $\boxed{10}$ Units/Hrs

	Mon	Tues	Wed	Thur	Fri	TOT
8:00	10	20	20	30	30	110
9:00	12	40	35	30	40	157
10:00	30	40	35	30	40	175
11:00	30	40	50	40	50	210
12:00	28	35	45	40	40	188
1:00	40	40	50	30	40	200
2:00	40	40	50	30	40	200
3:00	40	35	40	40	40	195
4:00	35	35	40	40	40	190
TOTAL	265	325	365	310	360	1625

Template: FIG918
Author: Don Beil

Figure 9-18. The numbers enclosed in boxes are variables that we provide to this template to prepare our PRODUCTION THIS WEEK and our FORECAST for next week. All other values are calculated by the template.

9-18, we can simply rearrange the axes of each of these reports and prepare both with the days of the week across the rows and the times down the columns.

Let's study the full template designed to prepare the two reports shown in Figure 9-18. The complete template is shown in Figure 9-19. The final report of the previous figure is marked. Also marked is the initial screen. Notice the location of the cursor as it appears when the template is loaded. It appears on the Monday

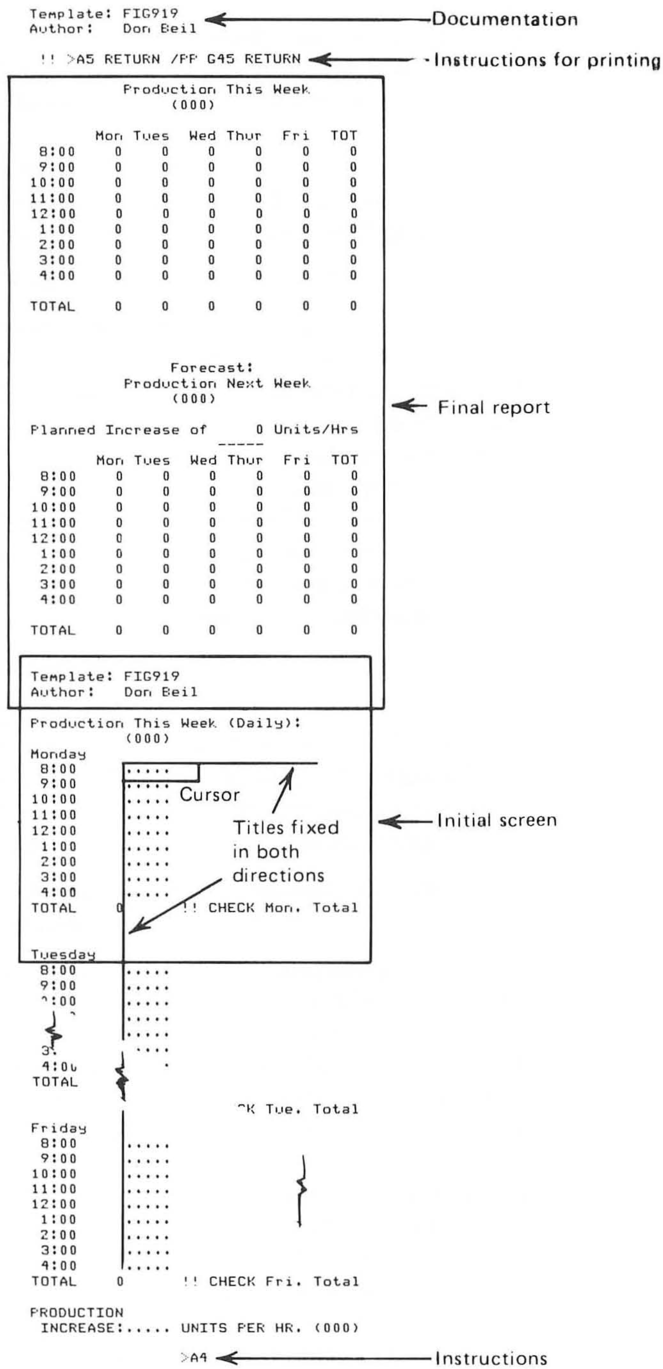


Figure 9-19. The full annotated template of our production example.

8:00 a.m. entry, ready for the first data to be entered. Titles have been fixed in both directions as noted, to limit access to the upper report area and to reduce the possibility of destroying formulas there. Each of these items is explained in earlier discussions in this chapter.

Look at the screen in Figure 9-20. We have just moved down the column entering the values for production this week on Monday. As we finish the 4:00 p.m. time, we come to row 60, which shows:

TOTAL 0 !! CHECK MON. TOTAL

This line contains an instruction, the two exclamation points (!!) that ask for a recalculation. This sheet has been stored with a global recalculation mode of manual (M), so that data could be entered more quickly. Now the user is asked to recalculate twice. Doing so provides the screen shown in Figure 9-21.

Notice the entry at location B60, which now shows 175 instead of 0. The recalculations have provided a Monday TOTAL in the data entry area of the sheet. This is a "protected" entry since it is behind (to the left of) the titles fixed in place on the template. The user is also instructed to

CHECK MON TOTAL

meaning that a total has been provided for Monday and that the individual can use that as a method of verifying that the numbers for Monday have been entered accurately. We'll assume that the user has calculated this value as a control total used to check on the accuracy of the process. We can list this idea as a guideline.

Template Guideline 15: Provide the user with aids to verify that data are being entered properly.

This is important, especially since VisiCalc has only limited techniques to edit user input to ascertain validity. For example, we cannot easily check a value entered to see if it is a number, not a label. We are also very limited in communicating errors or potential errors to users since we can't print messages to users from the sheet (except for the words ERROR and NA). Therefore, it is important to include whatever capability we can to ensure that data are entered correctly. In this example, we are assuming that the user has totaled the production for each day and that this total is compared against our built-in sum as a way of controlling the accuracy of the data. We want the user to check the data entered so that accurate information is prepared.

Notice with this template, that the user is required to enter data vertically down the sheet. This was done with reason.

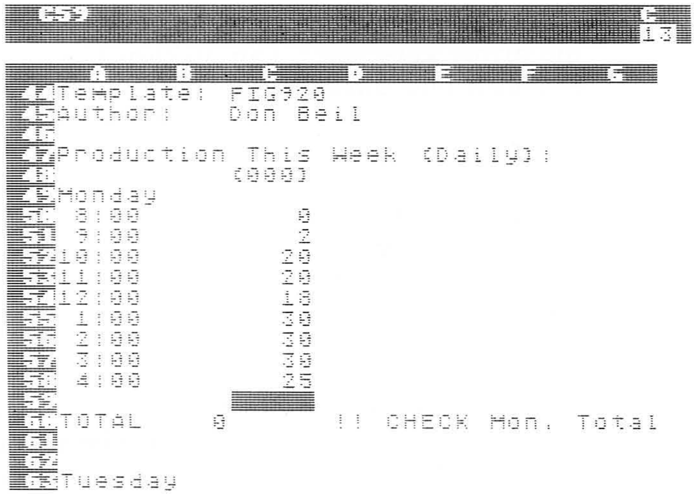


Figure 9-20. After moving down the column entering data, the user comes to an instruction to perform recalculations (!!).

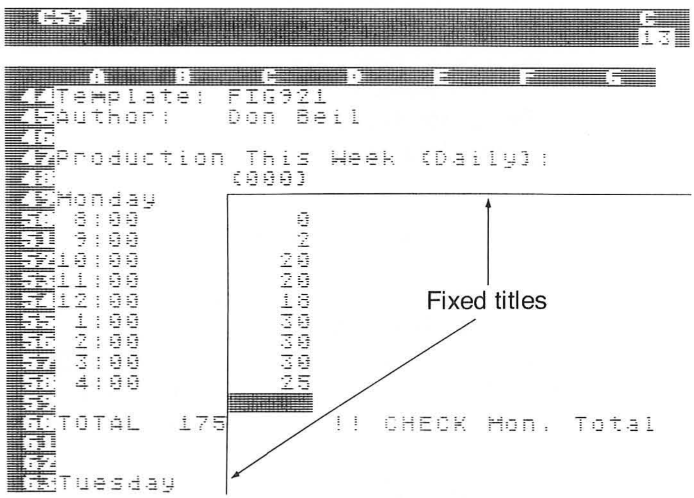


Figure 9-21. The recalculations provide a total for the day (here, 175) that gives the user a control total to be used to verify that accurate data have been entered.

Template Guideline 16: When building the data entry area of a template, require the user to enter data while moving down the template in a single column.

There are several reasons for this. First, moving the cursor with the arrow keys can be an awkward maneuver. Therefore, if we establish one direction (down) as the single movement required while entering data, we reduce the possibility of error. If large amounts of data must be entered, we can still go down the sheet until the bottom is reached. At that point, the user can be directed (with a GO TO) to the top of another column at which additional data are entered down the sheet. Second, more rows than columns appear on the screen, making it easier to follow the action down the sheet, rather than across. Finally, the movement seems less abrupt when scrolling down the sheet instead of across it.

In our template, the user continues down the screen until the data for each day of the week have been entered; then the desired PRODUCTION INCREASE is entered as prompted.

Continuing down the sheet, the user finds an instruction,

>A4.

When issued, the user is sent to line 4 of our template (refer to Figure 9-19). Here, the first instruction is to perform recalculations (!!). We'll review the reason next.

Template Guideline 17: Recalculate before printing.

Template Guideline 18: On spreadsheets requiring significant data entry, use the manual recalculation mode.

On this template, we placed the sheet into manual recalculation mode to speed the data entry process. Here, recalculations do occur after the data for each week are entered, but one is not included after the PRODUCTION INCREASE is entered at the bottom of the sheet. (We can, of course, include one if desired.) After the GO TO A4 is executed, our FORECAST table does not yet have the correct final values. The recalculation causes this. (Two exclamation points are not required, one is sufficient in this example.)

In Figure 9-22, the cursor is resting at position G19, where we have an incorrect entry. The formula at this entry, @SUM(B19...F19), was generated when a Replicate command was used to copy the formula down the G column from entry G10. As discussed earlier, in the Replicate section of Chapter 4, Commands, this can happen often with spreadsheet creation. One method to avoid it follows.

Template Guideline 19: Insert blank rows and columns as the final step of creating the template.

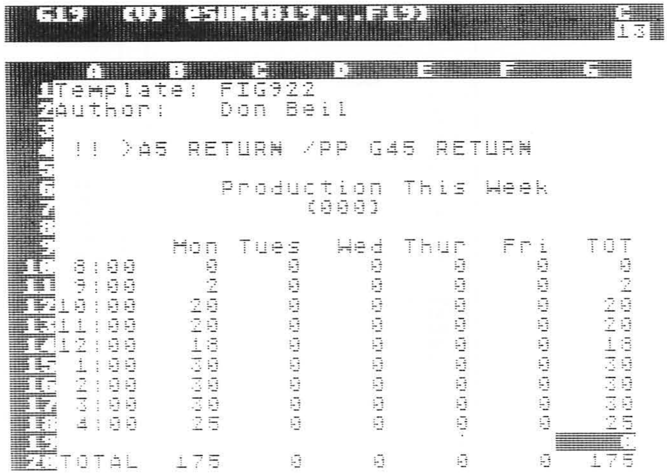


Figure 9-22. The cursor highlights an error that could have been avoided if this blank row had been entered after all other work on the template was complete.

Postponing this action can help avoid the work and potential problems associated with clearing individual entries that can occur when replicating as described.

Let's revise this template to illustrate another technique that can be used to write protect the data entered by the user. Figure 9-23 was formed by overlaying two VisiCalc sheets. One sheet, used for data entry, contains columns A through G. The second sheet, with the production report and forecast, contains columns I through O, and is blank in all other areas. The formulas of the sheet on the right refer to entries in column C.

To use these templates, we'll follow these steps.

Enter	Explanation
/C	Begin the Clear command.
Y	After confirming that we can clear the sheet, enter Y.
/SL FIG923L RETURN	Load the data entry area and then enter our production data and increase in column C.
/SL FIG923R RETURN	Overlay the report and forecast on the data. Since we do not want to destroy the data, we will not clear the sheet before this step.

A B C D E F G H I J K L M N O

```
Template: FIG923  left
Author:   Don Beil

Production This Week (Daily):
           (000)
Monday
 8:00       ....
 9:00       ....
10:00       ....
11:00       ....
12:00       ....
 1:00       ....
 2:00       ....
 3:00       ....
 4:00       ....

TOTAL    0       !! CHECK Mon. Total


Tuesday
 8:00      .....
 9:00      .....
10:00      .....
11:00      .....
12:00      .....
 1:00      .....
 2:00      .....
 3:00      .....
 4:00      .....
TOTAL    0       !! CHECK Tue. Total


Wednesday
 8:00      .....
 9:00      .....
10:00      .....
11:00      .....
12:00      .....
 1:00      .....
 2:00      .....
 3:00      .....
 4:00      .....
TOTAL    0       !! CHECK Wed. Total


Thursday
 8:00      .....
 9:00      .....
10:00      .....
11:00      .....
12:00      .....
 1:00      .....
 2:00      .....
 3:00      .....
 4:00      .....
TOTAL    0       !! CHECK Thur Total


Friday
 8:00      .....
 9:00      .....
10:00      .....
11:00      .....
12:00      .....
 1:00      .....
 2:00      .....
 3:00      .....
 4:00      .....
TOTAL    0       !! CHECK Fri. Total

PRODUCTION
 INCREASE:..... UNITS PER HR. (000)

                 >A4
```

```
Template: FIG923  right
Author:   Don Beil

 !! >A5 RETURN /PP G45 RETURN

         Production This Week
               (000)

      Mon  Tues  Wed  Thur  Fri  TOT
 8:00   0     0    0     0    0    0
 9:00   0     0    0     0    0    0
10:00   0     0    0     0    0    0
11:00   0     0    0     0    0    0
12:00   0     0    0     0    0    0
 1:00   0     0    0     0    0    0
 2:00   0     0    0     0    0    0
 3:00   0     0    0     0    0    0
 4:00   0     0    0     0    0    0

TOTAL   0     0    0     0    0    0



              Forecast:
         Production Next Week
               (000)

Planned Increase of      0 Units/Hrs
                     -----
      Mon  Tues  Wed  Thur  Fri  TOT
 8:00   0     0    0     0    0    0
 9:00   0     0    0     0    0    0
10:00   0     0    0     0    0    0
11:00   0     0    0     0    0    0
12:00   0     0    0     0    0    0
 1:00   0     0    0     0    0    0
 2:00   0     0    0     0    0    0
 3:00   0     0    0     0    0    0
 4:00   0     0    0     0    0    0

TOTAL   0     0    0     0    0    0
```

Figure 9-23. One spreadsheet resulting from overlaying a data entry matrix (columns A through G) on the desired model (columns I through O).

Figure 9-23. One spreadsheet resulting from overlaying a data entry matrix (columns A through G) on the desired model (columns I through O).

At this point, we have our desired information. The advantage of this technique is that it isolates data entry while placing the data entered to the left of the model. That is the desired location in recalculation since it allows the entered values to flow into the model correctly.

GUIDELINES FOR CREATING TEMPLATES: EXAMPLE 3

Our final example in the chapter will be a straightforward template used to estimate our electric utility bill per month for the year. We begin with records of the monthly usage last year which we enter as one column. Then we'll enter only two values:

- Percentage increase or decrease we anticipate in kilowatt usage for the year.
- Expected price.

We want to build an interactive template that allows us to change these two variables and observe the results of our actions. Figure 9-24 contains such a template. The opening screen contains two windows as shown. The two variables are entered in the lower window and the results are displayed in the upper window. As with other examples, titles are fixed as marked.

The split window also provides additional protection against destroying the upper segment of the screen since it can be used to display noncontiguous sections simultaneously, thus allowing us to see the data entry area and the resulting fields, which are safely separated on the actual sheet as shown.

> **Template Guideline 20:** Split windows can provide extra protection in preserving sections of the sheet from accidental destruction.

This sheet also contains several other interesting sections. Notice in the illustration of the full sheet of Figure 9-24 that the last line contains instructions to begin printing if desired. First, reduce the screen to one window (/W1), then GO TO A4. Then at location A4, we're told to

ENTER DATE AND TIME ON ROW 8.

Notice that row 8 contains

DATE: "MM/DD/YY TIME: "00:00

If we tried to enter either without a double quote ("), we wouldn't be able to enter the slash (/) which would cause division, or colon (:) since VisiCalc would assume that we are entering a value. Thus, the instructions shown with the quote will, if followed, cause VisiCalc to accept the two items in the correct formats.

```
Template FIG924
Author    Don Beil

Enter date and time on row  8,
then follow directions on row 6.
 >A8 RETURN /PP D34 RETURN

   Date: "MM/DD/YY    Time: "00:00

ESTIMATED KILOWATT USAGE AND COST:
```

	KWH Last YR	KWH This YR	COST This YR
January	500	500	0.00
February	500	500	0.00
March	500	500	0.00
April	500	500	0.00
May	500	500	0.00
June	600	600	0.00
July	800	800	0.00
August	700	700	0.00
September	400	400	0.00
October	500	500	0.00
November	500	500	0.00
December	500	500	0.00
TOTALS	6500	6500	0.00

```
Template FIG924
Author    Don Beil
```

```
Inc/Dec this year:......... percent.
Price this year:  .........
To print: /W1 >A4
```

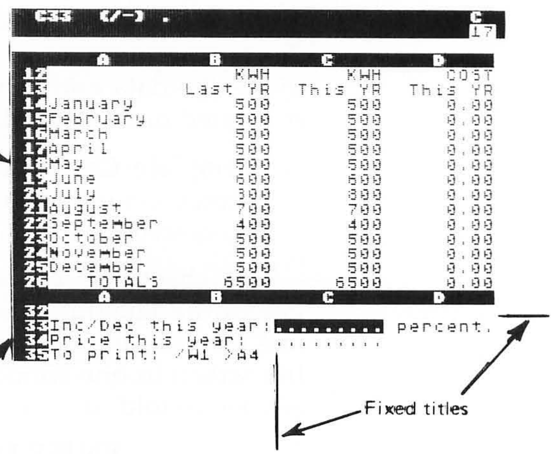


Figure 9-24. An interactive template that allows us to enter two variables and observe the impact on our electric consumption and cost.

> **Template Guideline 21:** Format the data entry location so that it will be easy to enter the data correctly.

Here we've included explicit directions ("MM/DD/YY) for how the data are to be entered and thus helped the user. At other times, we can indicate that values are to be entered in ways other than using repeated periods; for example, we can use VVVVVV, VVVV.VV, or VALUE. Similarly, areas to receive a label could be marked LLLLLL, LABEL, etc.

How do we know if our templates work correctly? This is not a simple question to answer. Refer to Chapter 8, Recognizing, Preventing, and Correcting Errors, for a discussion of this topic and for a number of suggestions applicable to this discussion on creating templates. Here, let's briefly summarize our earlier advice of that chapter.

> **Template Guideline 22:** Test templates before using them, check them while using them, and verify results after using them.

One of the major programming lessons of the last two decades has been to invest time and energy in designing a solution before beginning to code it. Coding is the process of actually writing and entering instructions. It should occur after we've mentally solved a problem and when we're ready to translate that solution to a format understandable to a computer system. This chapter began with an admonishment to understand the problem that we're solving. It ends with encouragement to understand the solution to be used before we begin to implement it.

Chapter 10

Documentation

INTRODUCTION

Documentation provides a record of our activities that can be helpful when we want to reuse our work, or can assist another person who uses spreadsheets that we prepare. Each of us must decide, using many factors, what is appropriate documentation for our efforts. Documentation of our work requires time and is often seen by those new to computers as unimportant; however, consider the importance of well-written manuals to guide in the use of computer hardware or accompanying software.

We hope the software that we use is well-documented both internally and externally. We'll explain both types in this chapter.

There are many examples of internal documentation. With VisiCalc this includes the information appearing on the prompt line to assist us in using VisiCalc correctly. As a specific example, when we enter the / (slash) to begin a command, the following appears on the prompt line:

COMMAND: BCDFGIMPRSTVW-

This documentation within the software makes it easier to use. VisiCalc contains a large number of these internal prompts.

The manual accompanying VisiCalc is the external documentation provided for it. It is separate from the software but important to our understanding of how the software functions.

As we create software, in the form of VisiCalc spreadsheets, we'll want to prepare both internal and external documentation. Let's discuss both kinds of documentation for our VisiCalc activities and suggest methods of preparing both.

PROVIDING INTERNAL DOCUMENTATION ON THE SPREADSHEET

When using VisiCalc, it is possible to include reference data, instructions, or other information within the sheet by entering this information as labels on the spreadsheet. One example is a suggested standard documentation format that could appear within every one of our electronic sheets. Figure 10-1 provides a simple template layout for such internal documentation. It has been established by setting a wide global column width and then entering the labels as shown. For this example, the width was set at 18 by typing

/GC18 RETURN

Figure 10-2 was built from Figure 10-1 to illustrate how we can complete the template for a specific spreadsheet.

Let's discuss the entries, momentarily ignoring the entry at location A1. On row 2 we've placed the file name of this spreadsheet (QTR4) and on row 3 the identification of the diskette (FORECAST-12). Rows 5 to 7 contain a brief description of the function of the electronic sheet. The next lines (rows 9 and 10) contain the name of the author of the sheet and the date written.

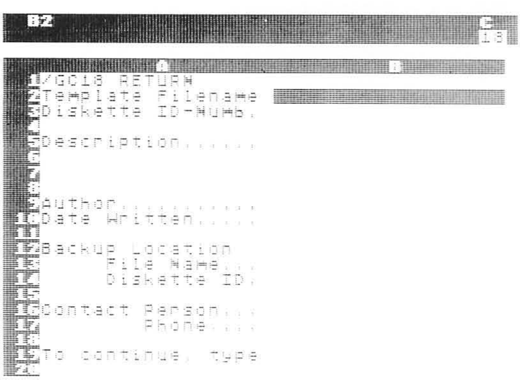


Figure 10-1. A template containing a format for internal documentation of a spreadsheet.

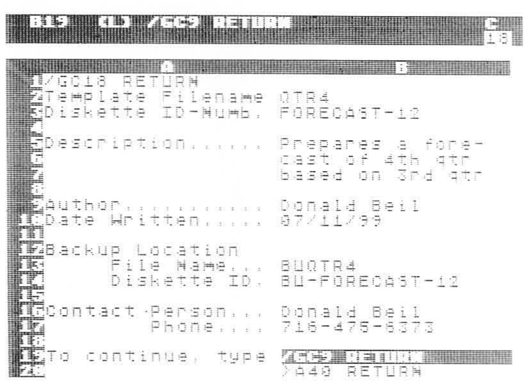


Figure 10-2. A specific example of our sheet with data for one of our budget spreadsheets.

These are followed by lines containing the location of the backup of this spreadsheet (the file and diskette names). As discussed earlier, the backup is a duplicate copy of this diskette (or one of several copies) retained in case this copy is inadvertently destroyed. The name and telephone number of the person to contact regarding the sheet follow.

Finally, there are directions to be followed to continue processing. In this example (Figure 10-2), the user is instructed to type

/GC9 RETURN
>A40 RETURN

The first of these commands reduces the column width to 9, which in this case is the width required for the remainder of the sheet. The next command, when entered, moves the cursor to position A40, where the user begins working with the sheet. Additional instructions then may be placed at that location. These commands have been written on the sheet by typing them as labels, beginning each with the double quote (") character.

Figure 10-3 displays the sheet after entering the first command to reduce the column width. It is confusing as shown; however, let's look again at position A1 of Figure 10-3. It shows

/GC18 RET

which, when entered, will restore this documentation page to its original appearance for the user. A person who wants to access

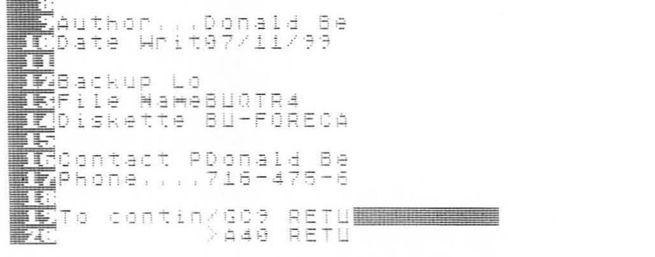


Figure 10-3. Our documentation window after reducing the column width.

this window of reference information while working on other areas of the sheet can GO TO A1 and enter the command found there to see the full page of internal documentation.

In this example, we also have several commands embedded on the sheet as directions. Let's look at an example of this in Figure 10-4. Notice that row 23 contains a series of instructions to be followed when it's time to print a report from this spreadsheet. Assume that we were sent to location A23 by a GO TO instruction contained elsewhere on the sheet. Upon arriving at this location, we'll follow the instructions to print this report on the available printer. Here, following the directions on line 23, we move the cursor to location A25, then enter the Print command as shown, and indicate H120 as the lower right corner of the report to be printed. This line can be extremely useful since it can reduce the searching for the lower right corner and the possibility for issuing an unwanted, and thus wasted, Print command.

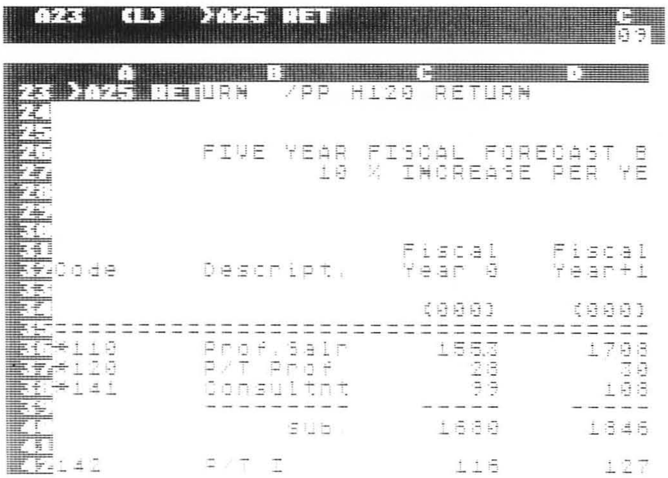


Figure 10-4. An example in which we've embedded instructions.

Figure 10-5 illustrates a more complicated example for the same report. Row 22 directs us to enter a date at location B28 by entering a double quotation mark and then the date in the format shown. This action will replace the guide

"MM/DD/YY

with the date that we enter.

Row 23 requests similar action for the time of day. Both the date and the time are commonly printed on computer reports and are important documentation elements. If many different reports are prepared, circulated, and filed from the same sheet, this action enables us to separate them based on date and time.

If we wish, we can insert instructions on the sheet directing the user to initiate a recalculation by entering an exclamation point, or we can include directions for how to save this sheet to a diskette or tape. These instructions can be included on the sheet in the same way that rows 22, 23, and 24 are included in the spreadsheet of Figure 10-5.

Finally, if we wish, we can include a full screen of documentation and directions on the spreadsheet or even have a full sheet consisting completely and only of documentation. To do so, we can establish the maximum column width available.

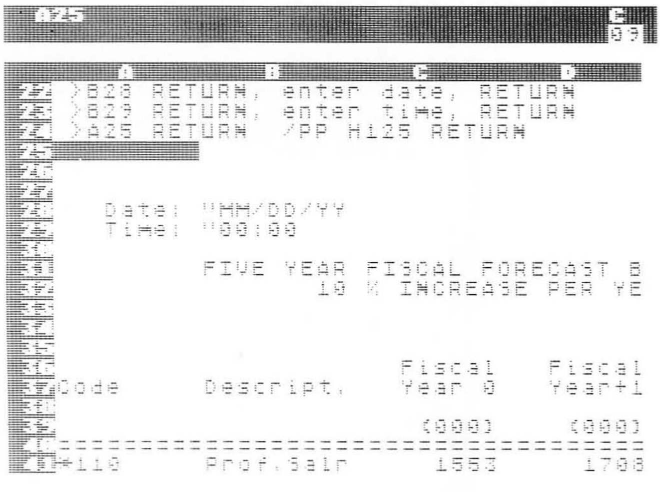


Figure 10-5. A window containing a series of embedded directions for the user of this spreadsheet.

PROVIDING EXTERNAL DOCUMENTATION FOR THE SPREADSHEET

In addition to the internal documentation that we develop, it's equally important to provide external documentation for the spreadsheet. In this section, we'll suggest information that could be provided to a user.

This additional documentation may consist of written pages including

- Sheet of directions and summary information.
- Sample copy of a report printed from the electronic sheet (if reports are printed).
- Sample copy of data collection forms on which the user records data or assumptions that will be used with this sheet.
- Listing that records usage of this spreadsheet (for example, the date used, the user name, files created, problems encountered).
- Other information needed, but not provided, on the spreadsheet, for example, a list of budget codes and their meanings as used on the template.
- Printed, annotated listing of the full template.

If documentation at this level is needed, then the use of simple plastic envelopes that can hold a diskette in one pocket, with accompanying documentation in another, may be helpful. They are available with 3-hole punches as shown in Figure 10-6 or designed to hang in a file cabinet.

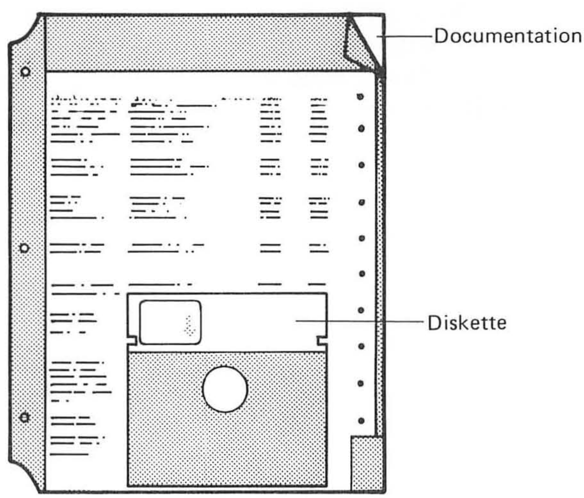


Figure 10-6. A convenient method of storing diskettes and documentation together.

Let's present a specific example of external documentation of a budget forecasting template. Page 1 of our documentation in Figure 10-7 contains a form on which we can record basic directions (as shown) as well as summary information for this particular spreadsheet.

Spreadsheet
Documentation

Page ______ of ______

Template Filename: ______________________________

Diskette Id-Number: ______________________________

Description: ______________________________

____

____

Author: ____________________

Date Written: ____________________

Back-up Location

Filename: ____________________

Diskette Id: ____________________

Contact Person: ____________________

Phone: ____________________

Note: The information above should also appear on the first window shown when loading the spreadsheet. Be certain that the correct sheet is being used by comparing the top two lines here with those on the window.

Directions:

USAGE RECORD

Date	User	New File Name	Notes, comments, problems

Figure 10-7. A sample documentation sheet for our templates, page 1.

Spreadsheet
Documentation

Page 1 of 4

Template Filename: FORE5YRS
Diskette Id-Number: FORECAST-11

Description: Prepares a report of projected vs. planned expenses.

Author: Donald Beil
Date Written: 04/23/99

Back-up Location
Filename: BU5YRS
Diskette Id: BU-FORECAST-11

Contact Person: Donald Beil
Phone: 716-475-6373

Note: The information above should also appear on the first window shown when loading the spreadsheet. Be certain that the correct sheet is being used by comparing the top two lines here with those on the window.

Directions:

1. Prepare input data on forms shown on page 3 of this documentation.
2. Boot VisiCalc
3. Insert diskette "FORECAST-11."
4. Load file FORE5YRS (/SL FORE5YRS RETURN)
5. Verify that correct diskette and file have been loaded.
6. Follow "continue" instructions on bottom of the window.
7. Enter data collected in step 1 above.
8. Follow directions to move cursor and print.
9. Distribute copies of this report to Departments 442, 481, and 530.

USAGE RECORD

Date	User	New File Name	Notes, comments, problems

Figure 10-7 continued.

Date: 06/15/99
Time: 12:45

FIVE YEAR FISCAL FORECAST BASED ON A
10 % INCREASE PER YEAR

Code	Descript.	Fiscal Year 0	Fiscal Year+1	Fiscal Year+2	Fiscal Year+3	Fiscal Year+4	Fiscal Year+5
		(000)	(000)	(000)	(000)	(000)	(000)
*110	Prof.Salr	1553	1708	1878	2065	2271	2498
*120	P/T Prof	28	30	33	36	39	42
*141	Consultnt	99	108	118	129	141	155
	sub.	1680	1846	2029	2230	2451	2695
142	P/T I	116	127	139	152	167	183
143	P/T II	54	59	64	70	77	84
	sub.	170	186	203	222	244	267
150	Hrly I	62	68	74	81	89	97
152	Hrly II	15	16	17	18	19	20
155	Hrly III	0	0	0	0	0	0
	sub.	77	84	91	99	108	117
*200	Benefits						
210	Soc Sec	209	229	251	276	303	333
220	Retirmt	112	123	135	148	162	178
232	L/T Dis	12	13	14	15	16	17
240	Hlth In	51	56	61	67	73	80
260	Maj Med	15	16	17	18	19	20
270	Wkm Cmp	27	29	31	34	37	40
	sub.	426	466	509	558	610	668
310	Suppls I	44	48	52	57	62	68
320	Suppls II	62	68	74	81	89	97
	sub.	106	116	126	138	151	165
408	Consltnts	72	79	86	94	103	113
590	Cons Trvl	12	13	14	15	16	17
	sub.	84	92	100	109	119	130
430	Telephone	25	27	29	31	34	37
	sub.	25	27	29	31	34	37
*755	Contngncy	10	11	12	13	14	15
756	Contng756	10	11	12	13	14	15
	sub.	20	22	24	26	28	30
**912	Equipment	120	132	145	159	174	191
**914	Equip>200	104	114	125	137	150	165
	sub.	224	246	270	296	324	356
Total $		777	854	939	1033	1136	1249

Figure 10-8. A sample report printed from our template. This report is page 2 of our documentation.

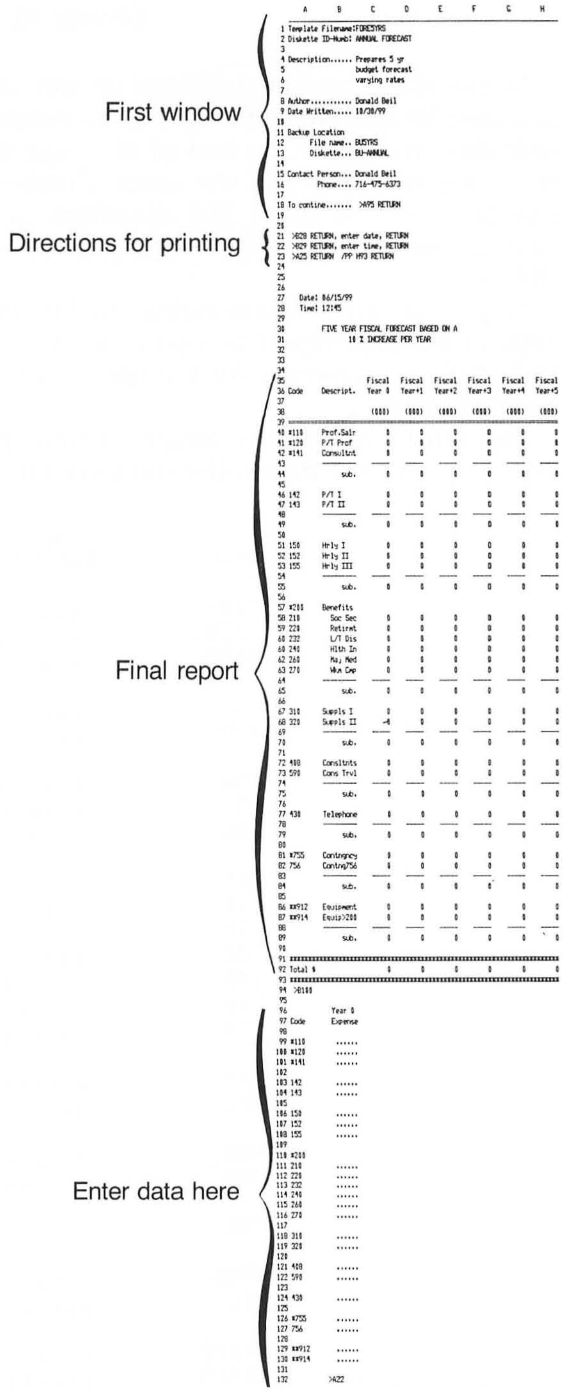
Notice the directions included on this page. They explain steps required by an individual to use the template successfully. It also contains an area at the end of the page that can serve as an activity log of the use of the sheet. Problems encountered, if any, can be recorded here. The directions also indicate procedures, such as step 9, to be followed to distribute printed reports from the sheet.

Page 2 of our documentation, in Figure 10-8, shows a sample copy of the final report printed from this spreadsheet. This page is included as a reference for the user to illustrate the expected final report format.

The third page of our documentation, page 3 shown in Figure 10-9, contains the data collection layout form used with this sheet.

Code	Year 0 Expense
*110	......
*120	......
*141	......
142	......
143	......
150	......
152	......
155	......
*200	
210	......
220	......
232	......
240	......
260	......
270	......
310	......
320	......
408	......
590	......
430	......
*755	......
756	......
**912	......
**914	......

Figure 10-9. A sheet that we can use to record the data to be used as input for this template, page 3 of our documentation.




```
          A      B      C      D      E      F      G      H
  1 Template Filename:FORESYRS
  2 Diskette ID-Numb: ANNUAL FORECAST
  3
  4 Description...... Prepares 5 yr
  5                   budget forecast
  6                   varying rates
  7
  8 Author........... Donald Beil
  9 Date Written..... 10/30/99
 10
 11 Backup Location
 12       File name.. BUSYRS
 13       Diskette... BU-ANNUAL
 14
 15 Contact Person... Donald Beil
 16         Phone.... 716-475-6373
 17
 18 To contine.......  >A95 RETURN
 19
 20
 21  >B28 RETURN, enter date, RETURN
 22  >B29 RETURN, enter time, RETURN
 23  >A25 RETURN  /PP H93 RETURN
 24
 25
 26
 27    Date: 06/15/99
 28    Time: 12:45
 29
 30          FIVE YEAR FISCAL FORECAST BASED ON A
 31                 10 % INCREASE PER YEAR
 32
 33
 34
 35                   Fiscal  Fiscal  Fiscal  Fiscal  Fiscal  Fiscal
 36 Code    Descript. Year 0  Year+1  Year+2  Year+3  Year+4  Year+5
 37
 38                   (000)   (000)   (000)   (000)   (000)   (000)
 39 ================================================================
 40 *110    Prof.Salr      0       0       0       0       0       0
 41 *120    P/T Prof       0       0       0       0       0       0
 42 *141    Consultnt      0       0       0       0       0       0
 43         --------- ------  ------  ------  ------  ------  ------
 44              sub.      0       0       0       0       0       0
 45
 46 142     P/T I          0       0       0       0       0       0
 47 143     P/T II         0       0       0       0       0       0
 48         --------- ------  ------  ------  ------  ------  ------
 49              sub.      0       0       0       0       0       0
 50
 51 150     Hrly I         0       0       0       0       0       0
 52 152     Hrly II        0       0       0       0       0       0
 53 155     Hrly III       0       0       0       0       0       0
 54         --------- ------  ------  ------  ------  ------  ------
 55              sub.      0       0       0       0       0       0
 56
 57 *200    Benefits
 58 210       Soc Sec      0       0       0       0       0       0
 59 220       Retirmt      0       0       0       0       0       0
 60 232       L/T Dis      0       0       0       0       0       0
 60 240       Hlth In      0       0       0       0       0       0
 62 260       Maj Med      0       0       0       0       0       0
 63 270       Wkn Cmp      0       0       0       0       0       0
 64         --------- ------  ------  ------  ------  ------  ------
 65              sub.      0       0       0       0       0       0
 66
 67 310     Suppls I       0       0       0       0       0       0
 68 320     Suppls II     -0       0       0       0       0       0
 69         --------- ------  ------  ------  ------  ------  ------
 70              sub.      0       0       0       0       0       0
 71
 72 400     Consltnts      0       0       0       0       0       0
 73 590     Cons Trvl      0       0       0       0       0       0
 74         --------- ------  ------  ------  ------  ------  ------
 75              sub.      0       0       0       0       0       0
 76
 77 430     Telephone      0       0       0       0       0       0
 78         --------- ------  ------  ------  ------  ------  ------
 79              sub.      0       0       0       0       0       0
 80
 81 *755    Contngncy      0       0       0       0       0       0
 82 756     Contng756      0       0       0       0       0       0
 83         --------- ------  ------  ------  ------  ------  ------
 84              sub.      0       0       0       0       0       0
 85
 86 **912   Equipment      0       0       0       0       0       0
 87 **914   Equip>200      0       0       0       0       0       0
 88         --------- ------  ------  ------  ------  ------  ------
 89              sub.      0       0       0       0       0       0
 90
 91 ################################################################
 92 Total $                0       0       0       0       0       0
 93 ################################################################
 94  >B100
 95
 96           Year 0
 97 Code      Expense
 98
 99 *110      ......
100 *120      ......
101 *141      ......
102
103 142       ......
104 143       ......
105
106 150       ......
107 152       ......
108 155       ......
109
110 *200
111 210       ......
112 220       ......
113 232       ......
114 240       ......
115 260       ......
116 270       ......
117
118 310       ......
119 320       ......
120
121 400       ......
122 590       ......
123
124 430       ......
125
126 *755      ......
127 756       ......
128
129 **912     ......
130 **914     ......
131
132          >A22
```

Figure 10-10. A printed list, page 4 of our documentation, of the full electronic sheet with written notes showing the relationships between sections of the sheet.

On it the user can write the assumptions (data) to be used in preparing this sheet. This page has been printed directly from the template itself. It is the portion of the sheet on which the user will enter the data for the sheet. Using this exact layout as a data collection and layout form provides an important connection between the sheet and the data entry function.

Finally, Figure 10-10 contains an annotated listing of the full sheet we have prepared. Notes written on this page explain it to the user; the entry coordinates of the sheet have been printed above (A, B, C,...) and to the left (1, 2, 3,...) of the listing. This page can be important in our training session with the user to explain the sheet fully.

SUMMARY

If a template will be used regularly, then attention to documentation when the sheet is created and first used will be of significant value later. As we'll see in Chapter 11, What Our Client, Secretary, or Supervisor Needs to Know, we must be prepared to train others who may use our sheets to understand and use our directions.

Let's emphasize that the length and format of the documentation may not be of importance; the quality and usefulness measure its value.

Chapter 11

What Our Client, Secretary, or Supervisor Needs To Know

How Others Use Our Templates Successfully

INTRODUCTION

This chapter is intended to be used by a knowledgeable VisiCalc user to train another person to use VisiCalc templates successfully. We'll assume that the trainee has no computer experience and that our training has a limited purpose: the trainee will learn how to use templates prepared by another person. It is not our intention here to teach an individual how to prepare templates, only how to use existing templates successfully.

However, this chapter could also be used as a starting point of more extensive training on the use of VisiCalc. An individual who's had the short training course of this chapter and who's gained experience from using our templates could be given this book for self-study on the full use of a VisiCalc System. In such an extended course, the Table of Contents of this book could serve as an outline, with this chapter used as a starting point.

This chapter is presented in outline form rather than narrative. As such, it provides a series of topics for a training session that we'll conduct. We'll elaborate on each item, or add items of our own, in meeting with trainees. The trainees can, of course, take notes on definitions, procedures, or other topics important to them.

A number of topics occur more than once on the outline since they are appropriate for discussion at several points, for general review, or for emphasis.

The training is intended to be performed at the computer with this outline in hand and with templates that the trainee will be expected to use. The trainee can be provided with the book for self study in getting started with our templates.

This training, and the communication established by it, is extremely important, for as we've recognized before, complete training of others who will use our templates is crucial to successful use of VisiCalc.

TRAINING SESSION: OUTLINE

On Using Existing VisiCalc Templates for Trainees Unfamiliar with Computers

I. Training Goals

A. Develop knowledge to use successfully VisiCalc templates written by another person.

B. Develop an understanding that VisiCalc itself is a productive tool only as part of a system that involves the individual users, the computer hardware, the templates used, the accompanying documentation, the data used, and the way the data are entered.

C. Develop communication between the author of the template and the trainees regarding procedures of use, actions to take if problems occur, and methods for the trainees to communicate suggestions for revisions or improvement in the VisiCalc system.

II. Hardware

(VisiCalc is used, or run, on a computer system; the components important for VisiCalc are discussed here.)

A. Communicating with the computer.
   1. Keyboard (input).
   2. Monitor or TV (output).
   3. Printer, if available (output).
   4. Other input or output devices.

B. Memory.

C. Disk drives and diskettes.

D. Cassette recorder (if used).

E. Turning the computer (and its components) on and off.

F. Basic care required.
   1. Respect for the hardware.
   2. Simple cautions (don't open the hardware, no food or drink nearby, etc.).
   3. Person to contact if trouble occurs.

III. VisiCalc

A. VisiCalc is a software program with an accompanying manual enabling us to use the hardware for specific purposes.

B. Loading VisiCalc.
   1. Procedures for loading this program into memory from a diskette.
   2. Cautions: Handle diskette carefully, avoid heat (do not place on top of equipment), avoid magnetic fields, do not bend, do not force into disk drive, load slowly and cautiously, handle the diskette only on the jacket, and do not touch the diskette surface.
   3. After loading, remove diskette, replacing it in its protective envelope and then in the binder.
   4. Successful loading results in a screen like that of Figure 11-1.

C. The Electronic sheet (refer to a live screen, or Figure 11-1).
   1. Similarities to large accounting worksheets that can be moved "under" the screen like a sheet of microfiche or can be viewed through the screen like with a magnifying glass.
   2. Columns, labeled A, B, C,...
   3. Rows, labeled 1, 2, 3,...
   4. Entries (intersection of columns and rows) are named A1, C30, G12, etc.
   5. Cursor.
      a. Defined.
      b. Moving the cursor.
         Arrow keys and the auto-repeat capability.
         GO TO.

D. Writing on the electronic sheet. This topic is presented under V. Entering Data.

Figure 11-1. The screen after VisiCalc has been successfully loaded into the memory of the computer (refer to Chapter 1, Introduction, for details of the specific hardware and VisiCalc version used in this book).

IV. Templates

(The trainer may wish to discuss this section while referring to an actual template that the trainee will be using.)

A. Templates—electronic sheets, like blank forms, on which labels, some values, and relationships between entries (formulas) have been placed. An individual uses them by entering values ("fill in the blanks") to obtain desired results.

B. Loading templates.
   1. First load VisiCalc (refer to III.B. of this outline).
   2. Procedures for loading a template from a diskette. (The documentation with the template may contain specific directions for this, as shown in steps 3, 4, and 5 of the directions portion of Figure 11-2.)
   3. Cautions with diskettes (refer to III.B.2. of this outline).

C. Template documentation.
   1. Review documentation format of existing templates.
      a. Documentation on the sheet itself (for example, refer to V.H.3. of this outline).
      b. Written documentation accompanying the template.

Spreadsheet
Documentation

Page ____ of ____

Template Filename: ______________________
Diskette Id-Number: ______________________

Description: ______________________
____
____

Author: ______________
Date Written: ______________

Back-up Location
Filename: ______________
Diskette Id: ______________

Contact Person: ______________
Phone: ______________

Note: The information above should also appear on the first window shown when loading the spreadsheet. Be certain that the correct sheet is being used by comparing the top two lines here with those on the window.

Directions:

USAGE RECORD

Date	User	New File Name	Notes, comments, problems

Figure 11-2. Part of the documentation prepared for a VisiCalc template.

Spreadsheet
Documentation

Page 1 of 4

Template Filename: FORE5YRS
Diskette Id-Number: FORECAST-11

Description: Prepares a report of projected vs. planned expenses.

Author: Donald Beil
Date Written: 04/23/99

Back-up Location
Filename: BU5YRS
Diskette Id: BU-FORECAST-11

Contact Person: Donald Beil
Phone: 716-475-6373

Note: The information above should also appear on the first window shown when loading the spreadsheet. Be certain that the correct sheet is being used by comparing the top two lines here with those on the window.

Directions:

1. Prepare input data on forms shown on page 3 of this documentation.
2. Boot VisiCalc
3. Insert diskette "FORECAST-11."
4. Load file FORE5YRS (/SL FORE5YRS RETURN)
5. Verify that correct diskette and file have been loaded.
6. Follow "continue" instructions on bottom of the window.
7. Enter data collected in step 1 above.
8. Follow directions to move cursor and print.
9. Distribute copies of this report to Departments 442, 481, and 530.

USAGE RECORD

Date	User	New File Name	Notes, comments, problems

Figure 11-2 continued.

2. Procedures when problems occur. Record problems on usage section of the documentation.

V. Entering Data

(The trainer may wish to discuss this section while referring to an actual template the trainee will be using.)

A. This is the process of communicating with the template and writing data onto the electronic sheet from a paper sheet or other source.

B. Keyboard.
   1. Letters. Availability of upper and lower case.
   2. Numbers.
   3. Special characters (. ! " * + -, etc.).
   4. Control keys (as available):
      Arrow keys
      Return
      Shift
      Escape
      Control
      BACK S
      BREAK
      System Reset
      > (GO TO)
      ;
      /
      Space bar

C. VisiCalc sheets (refer to III.C. of this outline).

D. Entering numbers.
   1. Commas and dollar signs are not acceptable. Therefore, $112.16 or 1,000 or $12,617.28 all are incorrect. Enter 112.16 or 1000 or 12617.28 instead.
   2. Leading or trailing zeros are acceptable but are ignored. Thus, 12.6 and 0012.6 and 12.60, etc., all are evaluated identically.
   3. A decimal point, positive or negative sign, and scientific notation (42E-10) are acceptable.

E. Entering labels.
   1. Availability of upper and lower case letters.
   2. Use of the double quote (") for entering items as prompted by the template, for example, "MM/DD/YY.

F. Correcting typing errors.
   1. The BACK S key.
   2. The BACK S key used with auto-repeat, and the BREAK key.
   3. If the return key has been entered, correct the error by simply reentering.

G. Cautions.
   1. System reset key.
   2. Keys that are often confused:
      1 (number) and I (letter).
      0 (number) and O (letter).
   3. "Beep" and flashing cursor occur:
      a. at edge of sheet.
      b. when bumping into fixed titles.
      c. when some erroneous keys are entered; for example, in attempting to enter 1047, a beep will occur if the letter O (oh) is pressed instead of the digit 0 (zero).
   4. When interrupted (telephone, coffee break, lunch), do not turn off or unplug the computer unless all work is completed (files saved or printed as required), because this will destroy all data in memory.

H. Printing (if appropriate).
   1. Perform a recalculation (!) as needed.
   2. Preparing the printer.
      a. Power on.
      b. Load and align paper.
      c. Set controls (if necessary).
   3. Follow directions on the template as appropriate. For example, if

      /PP J77 RETURN

      appears, enter this series of characters, ignoring the spaces, and pressing the return key where the word RETURN appears.
   4. Verify output as appropriate: was a full report (from top to bottom, left to right) printed? Do numbers, totals, etc., appear reasonable?
   5. Distribute copies of the report if necessary.

I. Storage and backup of files.
   1. Store templates with data entered on diskettes or

tape cassettes as directed. Changing the templates in memory does not alter the stored file; we must do that deliberately if required.
2. Handle diskettes (cassettes) with care (see III.B.2. and III.B.3.).
3. Prepare backup copies of the files as directed.

J. Other procedures.
1. Perform recalculations (!) as directed.
2. If the same template is to be used with multiple sets of data, consider clearing memory and reloading the template for each to prevent the possibility that a relationship (formula) inadvertently destroyed in one step of data entry will create errors in subsequent uses.
3. Steps to complete the use of a template.
   a. Power off.
   b. Record activity in usage record (log) of documentation.
   c. Store diskettes.

VI. Handling Problems

A. Screen shows ERROR unexpectedly or other problems.
1. Recalculate (!).
2. Verify that all required values have been entered.
3. Reload the sheet and enter all data again.

B. Machine inadvertently powered off, system reset key depressed in error.

C. Record problems in the usage record (log) of the documentation as required.

D. Name and telephone number of person to contact with questions or problems.

VII. Other Topics

A. Suggestions for improvement or revisions in the template, documentation, or procedures should be made to: person, phone number.

Chapter 12

The Limitations of a VisiCalc System

INTRODUCTION

Each of us might want to be aware of VisiCalc's limitations so that we'll know if VisiCalc can be used to solve the problems facing us. We'll find, however, that we'll need to consider more than simply this software. We'll need to think in terms of a computer system composed of

- VisiCalc and the electronic sheets we create.
- The hardware we use.
- The data we prepare.
- The users (ourselves or others).

Within this general framework let's develop limitations for such a system. They are interrelated; the discussion will not neatly divide into the following headings. In each case, we'll try to concentrate on the limitations from a perspective of a user of a VisiCalc System, which means the VisiCalc program plus the other components already listed.

In addition to the limitations below, the other chapters of this book also discuss limitations of these systems.

In examining these limitations, it should be clear that the success and extreme popularity of this product result from its *capabilities*. It is widely recognized as an innovative, important, problem solving tool. The large number of articles referenced in the Bibliography contain an extremely small number of criticisms of its performance. A reader evaluating this product needs to balance the contents of this chapter against the capabilities described throughout the remainder of this book.

Some of the VisiCalc system limitations are listed next. For each of us, they are limitations only if they interfere with our ability to solve the individual problems facing us. Some of the items listed could be described as desired enhancements.

Some limitations occur because of design decisions regarding the operational structure of the program. VisiCalc is loaded into memory from the program diskette and then that diskette is no longer needed. This design means that the system works without reading separate modules, as needed, from the program diskette. It is, therefore, more convenient and faster to use than a different design that might have additional capability at the expense of added memory or time. Additional capabilities could reduce the amount of memory available for our spreadsheets. If the design had been established to provide for module access by swapping between program and data diskettes, that would have required additional time to solve our problems.

VISICALC AND OUR ELECTRONIC SHEETS

Within this framework, let's examine some limitations of the VisiCalc software.

Individual entries cannot be "locked" to prevent the accidental destruction of formulas or data. This can be a significant limitation for some applications.

In some instances, the system of nested capabilities within commands means that we need to know where to start to accomplish some lower level operations. VisiCalc *is* well designed in this area, but each of us may have trouble remembering where particular operations are to be found.

There is only limited capability provided to manipulate character data (as opposed to the powerful capabilities with numeric data). For example, we can enter characters as labels but cannot process them. We cannot communicate well from our sheets to the person using them, except in awkward ways. We cannot print English messages when certain events occur.

A label cannot be copied from one location to another, although a value can be. For example, if we want to copy the numeric value from D30 to location F14, we can place (D30) at F14 and thus copy its value. But a label cannot be copied in this way. This can be a limitation during the data entry process with templates.

Entering labels that extend beyond one column is an awkward process. All columns must be the same width.

There are no "execute" capabilities provided directly. We cannot create a file of data and commands together and then run a VisiCalc sheet with the file. If one printer requires a relatively lengthy setup string, we cannot enter that string once and then recall it as needed. If we have a large sheet that we regularly print in parts, we cannot store and then execute the series of Print and GO TO commands we use for printing it. Instead we must enter these commands separately each time we need to print the sheet.

VisiCalc does not have sorting capabilities or sophisticated selection capabilities. We cannot get a list of the top salespeople from each department in descending order by their sales. However, there are companion products for VisiCalc which interface with it and provide these capabilities.

The limitation on significant digits that can be stored can cause a problem in some applications. There is only limited control on the number of decimal places displayed (two with /F$ or none with /FI). Dollar and cents formatting does not include capability for commas, for display of dollar signs, for an accounting-like display of negative numbers within parentheses, etc. There is no control on displays in scientific notation. There is only very limited graphing capability although powerful associated software that interfaces with VisiCalc is available.

VisiCalc does contain software problems, though they seem to be extremely few in number.

HARDWARE

The limitations of a VisiCalc system, when considered from a hardware perspective, are included in this section.

Computers have limited memory. We'll need to know if problems that we typically encounter will "fit" within the memory of our hardware system. This is a limitation for any computer system, for the problem that we want to solve may simply be too large for it. This is a limitation on both the individual problem that can be solved and on the role that VisiCalc can play in part of a larger system.

Using the directional arrows by pressing the CTRL key in connection with an arrow key can be awkward to use.

Only a limited number of characters from the sheet may be displayed horizontally, and only a limited number of rows may be displayed vertically. Only a small part of the largest potential sheet can be displayed. However, VisiCalc overcomes this well with the Window and Titles commands.

Three frequently used keys, BACK S, BREAK, and RETURN are close together on the keyboard and we can inadvertently press the incorrect key.

Because of the limited number of characters on most keyboards many keys must perform different functions depending on when they are entered. As an extreme example, the character R is used

- To initiate the Replicate command.
- To indicate relative when operating within the Replicate command.
- As an indicator of a row in several commands.
- To indicate right justification in the Format and Global Format commands.
- Within the Global command to change the recalculation mode.

The Print command can be awkward to use for some printers. We are not asked what printer we are using and we are not then prompted regarding carriage control, etc. Instead, we must learn and use what may be a strange string of characters that must be repeated each time we want to print. However, for a user with a single printer, the required operation may quickly become a habit. The variety of printers available and their separate methods of operation are in part responsible for this.

DATA WE PREPARE

The checking capabilities necessary to verify that accurate data have been provided are not fully present. For example, if a formula requires the entry of a numeric value and a label is inadvertently entered, the label will be evaluated as a number (zero) and results will be obtained although they will not be flagged as incorrect. The process of verifying data in many computer systems often occupies a major portion of user prepared software. This capability is only provided in very limited ways with VisiCalc.

The VisiCalc system may be difficult to use with data over time. Year-to-date totals, in week-after-week processing, can be awkward to accumulate. This is also true for many file processing systems where this is a complicated problem.

The data entry process is not distinct from other aspects of using VisiCalc (however, see the suggestions in Chapter 9, Creating Templates). This means that preparing trails of activity is difficult and also implies that more reliance is placed on the person doing data entry than on the prepared sheet since it's awkward to verify data entry activity as in other computer system.

USERS (OURSELVES OR OTHERS)

As with other computer systems the users are the crucial element. We must fully understand what we have established, must be aware of all of the limitations, and must regularly verify that results upon which we act and make decisions are indeed accurate. Regular vigilance is necessary with VisiCalc.

We must focus full attention on our task, especially data entry, because of the ease with which formula relationships can be destroyed. If we inadvertently enter a number over a formula, the formula is destroyed.

If others use our electronic sheets, they must be trained and their training must be more than cursory. They must understand the full system and their part within it. They must know how to handle problem situations. VisiCalc systems place heavy responsibilities on the users and these must be understood.

Chapter 13

Practice Problems

INTRODUCTION

This chapter presents a wide variety of problems that can be used for practice in developing skill with VisiCalc. Work with these problems may suggest new ways in which this software tool can help us solve problems that we encounter.

This chapter can also be used for an educational training activity in a company, a school, or a university or for a seminar or workshop on VisiCalc for which this book is used.

Finally, this chapter can serve as a source of pleasure for those who simply enjoy solving problems with VisiCalc.

The sheets that we prepare may be of value for later use and therefore may be of value to save.

PROBLEMS

1. Prepare a spreadsheet to be used to determine the total price of a microcomputer system. In one column list the name of each item of hardware and software to be purchased. In the next column list the price of each. Subtotal and total the sheet as shown in Figure 13-1.

2. Produce a template containing a documentation format that will serve as the standard beginning for all your template prepara-

```
                 Microcomputer Costs

Hardware                                 cost
     Microcomputer
     Disk Drive
     TV or Monitor
     Printer                           __________
          Subtotal

Software
     VisiCalc
     Word Processor
     Data Base                         __________
          Subtotal

          TOTAL
```

Figure 13-1. An electronic sheet that helps calculate the cost of a microcomputer system.

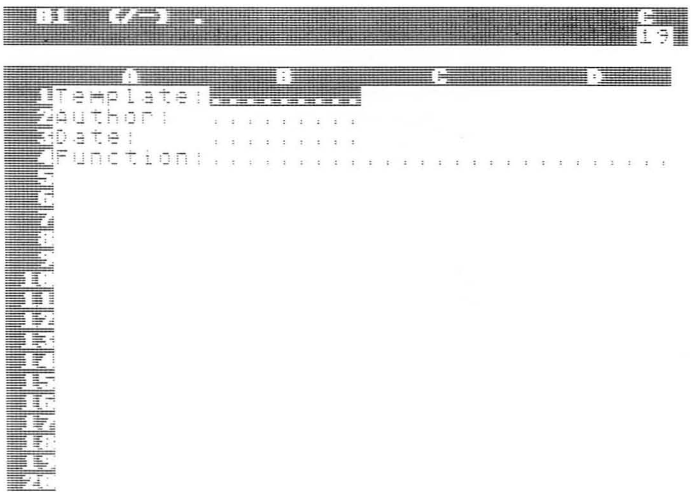


Figure 13-2. A suggested model template that can serve to start all our VisiCalc applications.

tion. Such a template might look like the one in Figure 13-2 and should be composed of data important for your environment.

3. Prepare a spreadsheet with the days of the week (Figure 13-3), another with the months of the year (Figure 13-4), and two others with weeks labeled from Week 1 across to Week 52 as shown in Figure 13-5 and Figure 13-6. For Figure 13-6 use the Rep-

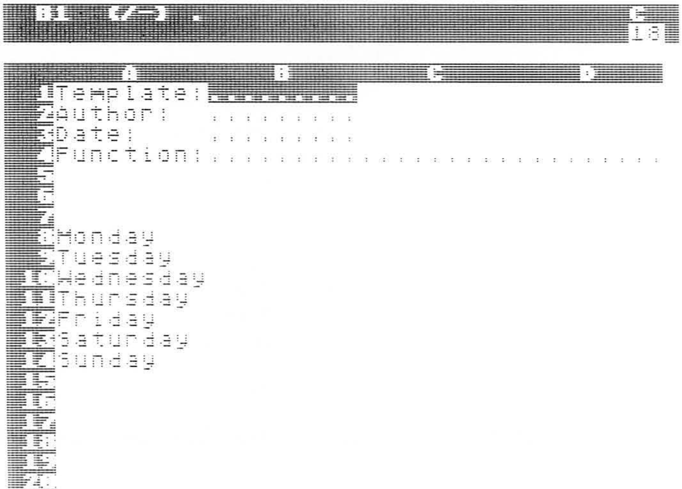


Figure 13-3. A starter template containing days of the week.

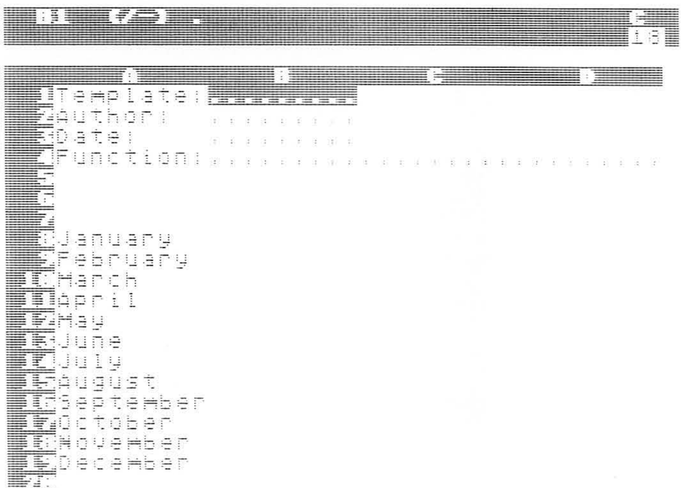


Figure 13-4. A template containing months of the year.

licate command for both rows. If you wish, prepare the 52-week template with the labels down the rows instead of across the columns as shown in Figure 13-7 and Figure 13-8. In Figure 13-8, right justify the word WEEK and left justify the number.

4. Prepare a spreadsheet, such as Figure 13-9, listing names of individuals. For your sheet, use the names shown, or use a set of

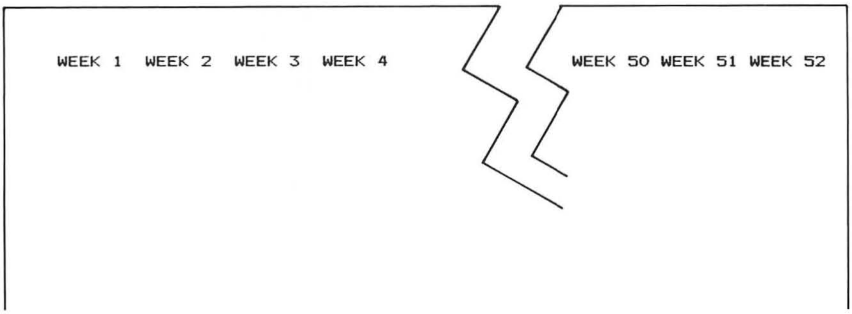


Figure 13-5. A template containing weeks of the year numbered horizontally across the sheet.

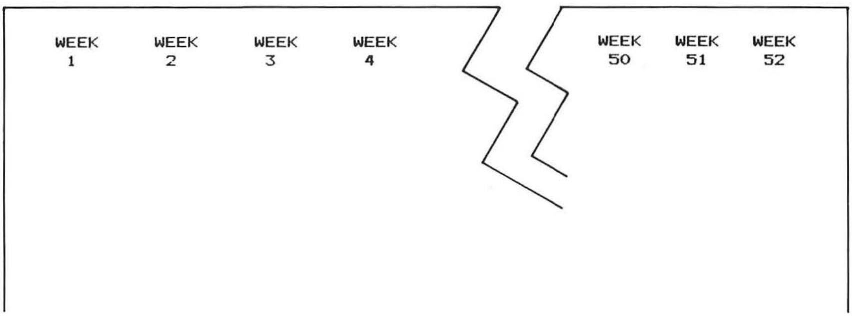


Figure 13-6. This template, containing weeks of the year numbered horizontally, can be prepared with the the Replicate command.

names important to you, for example, employees, customers, students, members of your family, etc. Store this sheet for possible later use in reports that you may need to prepare. If you report on activities by organizational unit, then establish a worksheet by unit such as shown in Figure 13-10.

VisiCalc does not have sorting capability; however, by inserting rows it is possible to prepare the report of Figure 13-9, beginning with an external unsorted list and sorting as the names are entered.

Prepare Figure 13-10 from Figure 13-9, using the Move command.

5. The salary and benefits expense codes, and their titles, for one organization are shown. Build a template which has this information. If you're the budget authority at your company or organi-

```
WEEK 1
WEEK 2
WEEK 3
WEEK 4
WEEK 5
WEEK 6
WEEK 7
WEEK 8
    .
    .
    .
WEEK 48
WEEK 49
WEEK 50
WEEK 51
WEEK 52
```

Figure 13-7. One version of a vertical presentation of the weeks of the year.

```
WEEK    1
WEEK    2
WEEK    3
WEEK    4
WEEK    5
WEEK    6
WEEK    7
WEEK    8
    .
    .
    .
WEEK   48
WEEK   49
WEEK   50
WEEK   51
WEEK   52
```

Figure 13-8. A second version of a vertical presentation of the weeks of the year.

zation, build this table from your codes and definitions. Build it once with columns 9 characters wide and again with columns 18 characters wide.

Code	Title
110	Salaries, administrative and professional
120	Salaries, other full-time
130	Salaries, technical, clerical and secretarial
141	Salaries, part-time, permanent admin/prof
142	Salaries, part-time, permanent genr'l/hourly

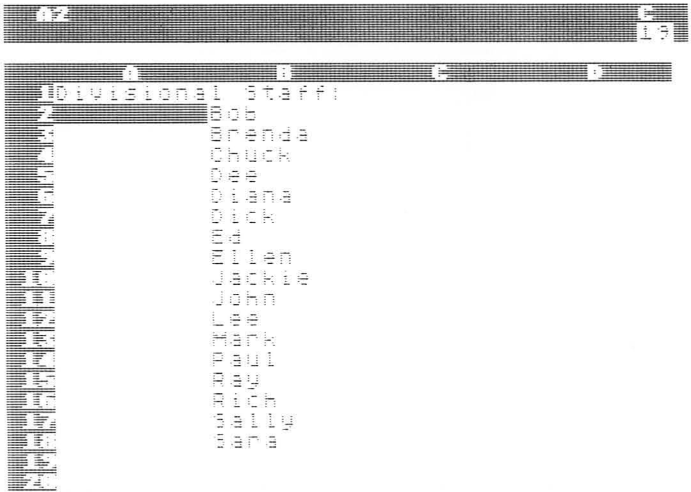


Figure 13-9. A prewritten template containing the names of divisional employees.

170	Salaries, other part-time, temporary
210	Social security expense
215	Unemployment insurance
220	Retirement contributions
230	Workmen's compensation
235	Disability benefits
240	Group health insurance
250	Group major medical
260	Group life insurance
270	Tuition

6. Prepare a spreadsheet with times of the day shown in intervals appropriate to a scheduling activity on which you prepare reports, for example, as in Figure 13-11.

7. On occasion it is desirable to be able to count columns on the worksheet, for example, to know how many columns there are from column M to column AR. Prepare a reference report like the one of Figure 13-12, which numbers each column identifier as shown. Right and left justify the data as shown in the columns. Change the column widths until you have a report that you feel is easy to read. This table can be used to compute "distances" between columns.

8. Prepare a spreadsheet that will provide you with a record of

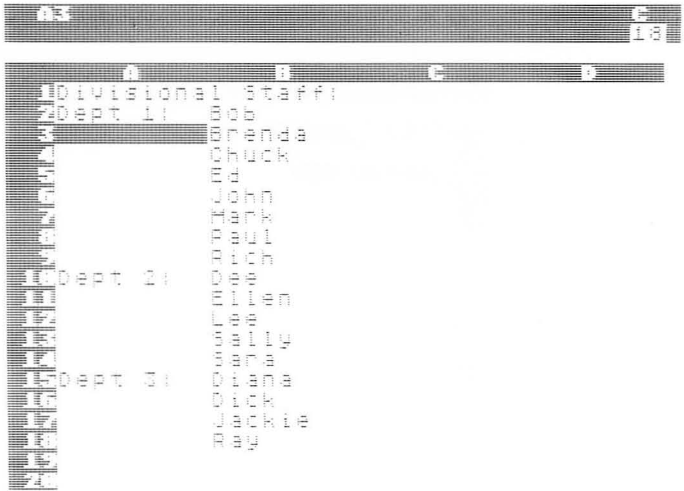


Figure 13-10. A file containing the names of divisional employees by department.

TIME

8:00
8:30
9:00
9:30
10:00
10:30
11:00
11:30
NOON
12:30
1:00
1:30
2:00
2:30
3:00
3:30
4:00
4:30
5:00
5:30
6:00
6:30
7:00
7:30
8:00
8:30
9:00
9:30
10:00
10:30

Figure 13-11. This sheet shows scheduled times of the day from which a report can be prepared.

A	1	AA	27	BA	53
B	2	AB	28	BB	54
C	3	AC	29	BC	55
D	4	AD	30	BD	56
E	5	AE	31	BE	57
F	6	AF	32	BF	58
G	7	AG	33	BG	59
H	8	AH	34	BH	60
I	9	AI	35	BI	61
J	10	AJ	36	BJ	62
K	11	AK	37	BK	63
L	12	AL	38		
M	13	AM	39		
N	14	AN	40		
O	15	AO	41		
P	16	AP	42		
Q	17	AQ	43		
R	18	AR	44		
S	19	AS	45		
T	20	AT	46		
U	21	AU	47		
V	22	AV	48		
W	23	AW	49		
X	24	AX	50		
Y	25	AY	51		
Z	26	AZ	52		

Figure 13-12. A reference table that can be used to "measure" the distance from one column to another.

your home or business consumption of electricity for a one-year period on a month-by-month basis. List the months down a column and then in another column record the consumption for each month in kilowatt-hours (kWh). Finally, add a third column that shows the year-to-date consumption. Include a total value for the second and third columns.

9. Develop a template on which you could record your consumption of natural gas for your business or home. Include report headings and then columns containing

- Month.
- Gas usage in hundreds of cubic feet (Ccf) used.
- Conversion to therms of the gas used, obtaining this value by multiplying Ccf by a 1.026 conversion factor.

Total the second and third columns.

10. Revise the last problem to include the three columns for last year, and then add two new columns for this year. These new columns should contain Ccf used and the conversion to therms. Finally add a column that calculates the difference between Ccf last year and this year.

11. Prepare a spreadsheet like the one in Figure 13-13, which contains a monthly record of your last year kilowatt (KWH) consumption, your forecast for this year, your actual consumption this year, and the difference to date between your forecast and actual usage. Your forecast for this year should be a variable that can be revised as shown by entering different values on the last line. Use the @NA value for coming months.

This problem demonstrates a forward reference. Save the sheet, then reload it, observing the need for a recalculation.

	KWH LAST YEAR	FORE- CAST KWH THIS YEAR	ACTUAL TO DATE	DIFF
JAN	800	720	700	20
FEB	850	765	650	115
MAR	920	828	700	128
APR	750	675	......	675
MAY	600	540	......	540
JUN	300	270	......	270
JUL	300	270	......	270
AUG	400	360	......	360
SEP	500	450	......	450
OCT	550	495	......	495
NOV	750	675	......	675
DEC	750	675	......	675
	7470	6723	2050	4673

FORECAST FOR THIS YR AT: -10 %

Figure 13-13. A record of actual and projected kilowatt usage.

12. A utility company allows customers to divide an estimated annual bill into equal amounts to be paid monthly. Prepare a spreadsheet, like the one in Figure 13-14, that can be used to compare actual against planned billing.

13. A large retail store maintains close records on weekly sales by department, such as the partial report shown in Figure 13-15. Complete the report for department A, B, C, D, and E for 52 weeks.

14. Figure 13-16 shows six weeks of a weekly sales report for one department of a retail store. Implement this report for 52 weeks.

15. A retail music store will be placing on sale all records whose

	EQUAL PAYMENT	ACTUAL BILL	DIFF. (EQUAL MINUS ACTUAL)
JAN	100.00	80.00	20.00
FEB	100.00	140.00	-40.00
MAR	100.00	95.00	5.00
APR	100.00	....	100.00
MAY	100.00	....	100.00
JUN	100.00	....	100.00
JUL	100.00	....	100.00
AUG	100.00	....	100.00
SEP	100.00	....	100.00
OCT	100.00	....	100.00
NOV	100.00	....	100.00
DEC	100.00	....	100.00
TOTAL	1200.00	315.00	885.00

Figure 13-14. This spreadsheet compares actual billing against a planned equal payment billing.

RETAIL SALES BY DEPARTMENT
LAST YEAR VS. THIS YEAR

WEEK		DEPT A	DEPT B	DEPT C
		------	------	------
1	LAST YR	14000		
	THIS YR	15100		
	DIFF:	1100		
2	LAST YR	850		
	THIS YR	400		
	DIFF:	-450		
3	LAST YR			
	THIS YR			
	DIFF:			

Figure 13-15. The beginning of a departmental comparison of retail sales this year against last year.

GROSS RETAIL SALES FOR:
DEPARTMENT: ** A **
BY WEEK

WEEK	LAST YR	THIS YR	DIFF. THIS WEEK	YEARLY SALES TO DATE --- YEAR ---- LAST	THIS	TO DATE DIFFERENCE $	%
				-----	-----	-----	-----
1	14000	15100	1100	14000	15100	1100	8
2	1000	800	-200	15000	15900	900	6
3	2000	2000	1000	17000	17900	900	5
4	0	1000	1000	17000	18900	1900	11
5	500	100	-400	17500	19000	1500	9
6	20300	21400	1100	37800	40400	2600	7

Figure 13-16. Weekly departmental sales report.

original selling price was from $5 to $10. The store is planning a sale at 80% of the original price. Prepare a table like that in Figure 13-17, which can be posted and placed by the cash register to display the sale price. Print the dollar sign ($) as shown.

16. Expand the spreadsheet of the last problem so that it calculates tax at 7% and then the total price, all prepared as in Figure 13-18.

17. Revise problem 15 with the Replicate command, so that the information is presented in four columns, not two.

18. Prepare an energy audit report template that can be used by a utility company to prepare a summary of savings and payback from various insulating activities. Use Figure 13-19 as your guide.

19. Prepare a template, such as the one in Figure 13-20, that can be used by a homeowner to estimate the average monthly consumption and cost of an electrical appliance. In the third column place a 1 (one) or a 0 (zero), with

value	meaning
1	in the home
0	not in the home

Then in the fourth column multiply column 2 by the cost per kWh (entered at the top of the report) and by the number in column 3 (either a zero or a one). This will place the approximate cost per month in the fourth column if the appliance is in the home and will place zero in the fourth column if it is not in the home. Compute yearly costs for each item, and total the monthly and yearly costs.

20. Develop a template that you can use to record individual medical expenses for family members. For every expense include the date, a description, the total cost, insurance payment if any, and net amount (total cost minus insurance payment). Subtotal for each family member and prepare a grand total. Establish subtotals so that new expeditures can be added without completely revising the formulas.

21. An employer prepares an annual summary of benefits for each employee. Prepare a single template to be used repeatedly for each individual with areas on which to enter a dollar value for each benefit. The benefits are group life insurance, long-term disability insurance, Social Security (FICA), retirement and annuity plan, health care, other benefits, and gross salary. Include a grand total.

ORIGINAL SELLING PRICE	80 % "SALE" PRICE
$ 5.00	$ 4.00
$ 5.05	$ 4.04
$ 5.10	$ 4.08
$ 5.15	$ 4.12
$ 5.20	$ 4.16
$ 5.25	$ 4.20
$ 5.30	$ 4.24
$ 5.35	$ 4.28
$ 5.40	$ 4.32
$ 5.45	$ 4.36
$ 5.50	$ 4.40
.	.
.	.
.	.
$ 9.75	$ 7.80
$ 9.80	$ 7.84
$ 9.85	$ 7.88
$ 9.90	$ 7.92
$ 9.95	$ 7.96
$ 10.00	$ 8.00

Figure 13-17. Part of a chart of original and sale prices.

ORIGINAL SELLING PRICE	80 % "SALE" PRICE	TAX 7%	TOTAL PRICE
$ 5.00	4.00	0.28	$ 4.28
$ 5.05	4.04	0.28	$ 4.32
$ 5.10	4.08	0.29	$ 4.37
$ 5.15	4.12	0.29	$ 4.41
$ 5.20	4.16	0.29	$ 4.45
$ 5.25	4.20	0.29	$ 4.49
$ 5.30	4.24	0.30	$ 4.54
$ 5.35	4.28	0.30	$ 4.58
$ 5.40	4.32	0.30	$ 4.62
$ 5.45	4.36	0.31	$ 4.67
.	.	.	.
.	.	.	.
.	.	.	.
$ 9.75	7.80	0.55	$ 8.35
$ 9.80	7.84	0.55	$ 8.39
$ 9.85	7.88	0.55	$ 8.43
$ 9.90	7.92	0.55	$ 8.47
$ 9.95	7.96	0.56	$ 8.52
$ 10.00	8.00	0.56	$ 8.56

Figure 13-18. An expanded sales price table that includes tax and total price.

ENERGY AUDIT

Name ______________________________
Address ____________________________
Town/City __________________________
Date ________________

Summary of Savings and payback period	Est. Cost	Yearly Savings	Payback* Years
A. Ceiling or Attic Insulation......	______	______	______
B. Wall Insulation..................	______	______	______
C. Storm Windows....................	______	______	______
D. Storm Doors......................	______	______	______
E. Weather Stripping................	______	______	______
F. Caulking.........................	______	______	______
G. Basement Wall Insulation.........	______	______	______
H. Floor Insulation.................	______	______	______
I. SUBTOTAL.........................	______	______	______
J. Automatic Thermostat.............	______	______	______
K. Water Heater Insulation..........	______	______	______
TOTAL............................	______	______	______

*Estimated Cost divided by Yearly Savings equals Payback Years.

Figure 13-19. A form from which an energy audit template can be prepared.

22. A basic record keeping system for shares of stock owned includes stock name, date purchased, number of shares, certificate number, price per share, stock cost, purchase fees, total cost, date sold, number of shares sold, selling price per share, total selling price, sales fees, net received, and profit/loss. Create a template with columns for each of these, with totals in appropriate columns.

23. A teacher must record grades for each student in a class on four examinations. Prepare a template with columns for student name, for the scores on each test, for the total, and for the percentage.

Electrical Consumption and Cost
(Excluding Heat and Lighting)

Enter cost per KWH:

Appliance	Approx. kWh/mo.	Used? Yes = 1 No = 0	Approx. Cost/ Mo.	Approx. Cost/ Yr.
FOOD PREPARATION:				
Microwave Oven	16			
Electric Range/oven	58			
Dishwasher	30			
Roaster	5			
Slow Cooker	11			
Toaster Oven	7			
Toaster	3			
Baby Food Warmer	2			
Electric Fry Pan	8			
SUB: Food Preparation:				
FOOD PRESERVATION:				
Freezer (manual defrost)	100			
100 Freezer (auto. defrost)	115			
Refrig-Freezer (manual defrost)	125			
Refrig-Freezer (auto defrost)	188			
SUB: Food Preservation				
LAUNDRY:				
Clothes Dryer (electric)	82			
Washer (automatic)	9			
Iron (hand)	5			
SUB: Laundry				
HEALTH AND COMFORT:				
Air cleaner (electronic)	43			
Air Conditioner (room)	71			
Dehumidifier (300 watts)	150			
Fan, Furnace	125			
Water Bed (heater)	150			
Water Heater (80 gal.)	600			
SUB: Health & Comfort				
HOME ENTERTAINMENT:				
Radio	7			
Radio/Record Player	9			
Television				
(b&w tube)	18			
(b&w solid state)	8			
(color, tube)	44			
(color, solid state)	26			
Pool filter	360			
SUB: Home Entertainment				
TOTALS				

Figure 13-20. A template design that can be used to estimate electrical consumption from home appliances.

24. Revise the spreadsheet from the previous problem to display the high and low score (use @MAX and @MIN) and the average (use @AVERAGE) for each test.

25. Use the @LOOKUP function to build a table of months of the year (1, 2, ... 12) and which can be used to look up the number of days in each month.

26. The market value of a treasury bill is computed by subtracting the discount from the face value. The discount is the face value times the number of days to maturity times the rate (which is the amount bid expressed as a decimal) divided by 360. Prepare an electronic sheet that accepts the variables in this formula and produces the discount and market value.

27. Some people say that a rough rule of thumb in computing life insurance and other death benefits is that they should total at least six times annual salary. Create the template shown in Figure 13-21 to assist in computing these values for a husband and wife.

28. Equity holdings are often shown in reports clustered by industry group. Figure 13-22 contains the beginnings of such a report. Using a similar report from a fund or other institution prepare a spreadsheet to track the holdings. The sheet should include a separate sum for each industry group and should show a group subtotal and the percentage (to two decimal places) that this group represents of the total holdings. The sheet should compute market value as the product of the number of shares and the price per share. The price per share should be on the sheet but should not be printed.

29. A realtor maintains records of the homes sold by the company in the city. For comparison purposes, data are maintained for the current year and last year on a weekly basis. Prepare a spreadsheet with columns for the two years and one row for each of the 52 weeks of the year. For each week list the number of homes sold, their total price (enter this as an expression, for example, 70000+112000+29000), the average price, and the year-to-date sales. Include these columns for this year and last year.

Develop an end-of-year procedure to "roll" the report from one year to the next. Basically all of the columns from this year should become comparison data, that is, data for last year.

30. Individuals with sole proprietorships use tax Schedule C of Form 1040, shown in Figure 13-23, to report business profit or loss. Set up a template in a format equivalent to the deductions portion of that form.

```
                    Life Insurance
             "Six-times-Salary" Formula.

                                              HUSBAND          WIFE
                                             _________       _______

1.  Enter present annual salary              $_____          $_____
2.  Multiply by six                             x6              x6
3.  Minimum protection goal                  $_____          $_____
4.  Enter coverage now held:
    (a) Personal policies            $_____          $_____
    (b) Group insurance               _____           _____
    (c) Pension plan death benefits   _____           _____
5.  Subtract total of (a) (b) (c)           -$_____         -$_____
6.  Additional insurance needed (if any)      _____           _____
```

Figure 13-21. Assumptions for estimating adequacy of life insurance holdings.

```
              Holdings By Industry Group

                                 Market
                       Shares    Value      Percent
                       ______    ______     _______
Aerospace:
     Stock 1            2040     125756
     Stock 2            8685     223641
     Stock 3            3667     403430
                                 752827       4.88%

Aluminum (and non-ferrous metals)
     Stock 1              .         .           .
        .                 .         .           .
        .                 .         .           .
```

Figure 13-22. The beginning of an equity holding report that shows diversification among industry groups.

31. Revise the previous problem so that the template is enlarged to have a separate area in which monthly expenses are recorded during the year in a column for each of the items listed. The yearly totals from each column total should then be automatically moved into the deductions format of the template.

32. Prepare a template that will complete Form 1040A of the U.S. Individual Income Tax Return shown in Figure 13-24. Assume that numeric values from other forms or from a table are entered here by hand. Include line number, brief description, and then columns as needed for the form.

SCHEDULE C
(Form 1040)
Department of the Treasury
Internal Revenue Service

Profit or (Loss) From Business or Profession

(Sole Proprietorship)

Partnerships, Joint Ventures, etc., Must File Form 1065.

▶ Attach to Form 1040 or Form 1041. ▶ See Instructions for Schedule C (Form 1040).

Name of proprietor | Social security number of proprietor

A Main business activity (see Instructions) ▶ ; product ▶

B Business name ▶ | C Employer identification number

D Business address (number and street) ▶

City, State and ZIP Code ▶

E Accounting method: (1) ☐ Cash (2) ☐ Accrual (3) ☐ Other (specify) ▶

F Method(s) used to value closing inventory:

(1) ☐ Cost (2) ☐ Lower of cost or market (3) ☐ Other (if other, attach explanation)

	Yes	No
G Was there any major change in determining quantities, costs, or valuations between opening and closing inventory? If "Yes," attach explanation.		
H Did you deduct expenses for an office in your home?		
I Did you elect to claim amortization (under section 191) or depreciation (under section 167(o)) for a rehabilitated certified historic structure (see Instructions)? (Amortizable basis (see Instructions) ▶)		

Part I Income

1 a Gross receipts or sales	1a	
b Returns and allowances	1b	
c Balance (subtract line 1b from line 1a)	1c	
2 Cost of goods sold and/or operations (Schedule C–1, line 8)	2	
3 Gross profit (subtract line 2 from line 1c)	3	
4 Other income (attach schedule)	4	
5 Total income (add lines 3 and 4) ▶	5	

Part II Deductions

6 Advertising		31 a Wages	
7 Amortization		b Jobs credit	
8 Bad debts from sales or services		c WIN credit	
9 Bank charges		d Total credits	
10 Car and truck expenses		e Subtract line 31d from 31a	
11 Commissions		32 Other expenses (specify):	
12 Depletion		a	
13 Depreciation (explain in Schedule C–2)		b	
14 Dues and publications		c	
15 Employee benefit programs		d	
16 Freight (not included on Schedule C–1)		e	
17 Insurance		f	
18 Interest on business indebtedness		g	
19 Laundry and cleaning		h	
20 Legal and professional services		i	
21 Office supplies		j	
22 Pension and profit-sharing plans		k	
23 Postage		l	
24 Rent on business property		m	
25 Repairs		n	
26 Supplies (not included on Schedule C–1)		o	
27 Taxes		p	
28 Telephone		q	
29 Travel and entertainment		r	
30 Utilities		s	

33 **Total deductions** (add amounts in columns for lines 6 through 32s) ▶ 33

34 Net profit or (loss) (subtract line 33 from line 5). If a profit, enter on Form 1040, line 13, and on Schedule SE, Part II, line 5a (or Form 1041, line 6). If a loss, go on to line 35 — 34

35 If you have a loss, do you have amounts for which you are not "at risk" in this business (see Instructions)? ☐ Yes ☐ No

Figure 13-23. IRS Form 1040, Schedule C.

Form **1040A** Department of the Treasury—Internal Revenue Service
U.S. Individual Income Tax Return

Use IRS label. Otherwise, please print or type.	Your first name and initial (if joint return, also give spouse's name and initial)	Last name	Your social security number
	Present home address (Number and street, including apartment number, or rural route)		Spouse's social security no.
	City, town or post office, State and ZIP code	Your occupation ▶	
		Spouse's occupation ▶	

Presidential Election Campaign Fund ▶ Do you want $1 to go to this fund? . . . . ☐ Yes ☐ No
If joint return, does your spouse want $1 to go to this fund? . . . ☐ Yes ☐ No
Note: *Checking "Yes" will not increase your tax or reduce your refund.*

Requested by Census Bureau for Revenue Sharing ▶
A Where do you live (actual location of residence)? (See page 6 of Instructions.) State | City, village, borough, etc.
B Do you live within the legal limits of a city, village, etc.? ☐ Yes ☐ No
C In what county do you live?
D In what township do you live?

For Privacy Act Notice, see page 27 of Instructions

For IRS use only

Filing Status
Check Only One Box.
1 ☐ Single
2 ☐ Married filing joint return (even if only one had income)
3 ☐ Married filing separate return. Enter spouse's social security no. above and full name here ▶
4 ☐ Head of household. (See pages 7 and 8 of Instructions.) If qualifying person is your unmarried child, enter child's name ▶

Exemptions
Always check the box labeled Yourself. Check other boxes if they apply.
5a ☐ Yourself ☐ 65 or over ☐ Blind
b ☐ Spouse ☐ 65 or over ☐ Blind
Enter number of boxes checked on 5a and b ▶ ☐
c First names of your dependent children who lived with you ▶
Enter number of children listed on 5c ▶ ☐

d Other dependents: (1) Name	(2) Relationship	(3) Number of months lived in your home.	(4) Did dependent have income of $1,000 or more?	(5) Did you provide more than one-half of dependent's support?

Enter number of other dependents ▶ ☐
6 Total number of exemptions claimed . . . . Add numbers entered in boxes above ▶ ☐

Please Attach Copy B of Forms W-2 Here

7 Wages, salaries, tips, etc. *(Attach Forms W-2. See page 10 of Instructions)* . . . . | 7
8 Interest income *(See pages 3 and 10 of Instructions)* . . . . | 8
9a Dividends .... *(See pages 3 and 10 of Instructions)* **9b** Exclusion .... Subtract line 9b from 9a | 9c
10a Unemployment compensation (insurance). Total received from Form(s) 1099–UC ....
b Taxable amount, if any, from worksheet on page 10 of Instructions . . . . | 10b
11 Adjusted gross income *(add lines 7, 8, 9c, and 10b). If under $10,000, see page 12 of Instructions on "Earned Income Credit"* . . . . | 11
12a Credit for contributions to candidates for public office. *(See page 11 of Instructions)* . . . . | 12a
IF YOU WANT IRS TO FIGURE YOUR TAX, PLEASE STOP HERE AND SIGN BELOW.
b Total Federal income tax withheld *(If line 7 is more than $25,900, see page 11 of Instructions)* . . . . | 12b
c Earned income credit *(from page 12 of Instructions)* . . . | 12c
13 Total *(add lines 12a, b, and c)* . . . . | 13
14a Tax on the amount on line 11. *(See page 13 of Instructions; then find your tax in the Tax Tables on pages 15–26)* . . . . | 14a
b Advance earned income credit (EIC) *(from Form W-2)* . . . | 14b
15 Total *(add lines 14a and 14b)* . . . . | 15
16 If line 13 is larger than line 15, enter amount to be **REFUNDED TO YOU** . . . . ▶ | 16
17 If line 15 is larger than line 13, enter **BALANCE DUE.** Attach check or money order for full amount payable to "Internal Revenue Service." Write your social security number on check or money order . ▶ | 17

Attach Payment Here

Please Sign Here
Under penalties of perjury, I declare that I have examined this return, including accompanying schedules and statements, and to the best of my knowledge and belief, it is true, correct, and complete. Declaration of preparer (other than taxpayer) is based on all information of which preparer has any knowledge.
▶ Your signature Date ▶ Spouse's signature (If filing jointly, BOTH must sign even if only one had income)

Paid Preparer's Use Only
Preparer's signature and date ▶ Check if self-employed ▶ ☐ Preparer's social security no.
Firm's name (or yours, if self-employed) and address ▶ E.I. No. ▶ ZIP code ▶

☆ U.S. GOVERNMENT PRINTING OFFICE: 1980—O-313-267 13-2687299
Form **1040A** (1980)

Figure 13-24. IRS Form 1040A.

Form **1040** Department of the Treasury—Internal Revenue Service
U.S. Individual Income Tax Return

For Privacy Act Notice, see Instructions | For the year January 1–December 31, 1980, or other tax year beginning , 1980, ending , 19

Use IRS label. Otherwise, please print or type.	Your first name and initial (if joint return, also give spouse's name and initial)	Last name	Your social security number
	Present home address (Number and street, including apartment number, or rural route)		Spouse's social security no.
	City, town or post office, State and ZIP code	Your occupation ▶	
		Spouse's occupation ▶	

Presidential Election Campaign Fund ▶ Do you want $1 to go to this fund? . . . . . . Yes No
If joint return, does your spouse want $1 to go to this fund? . . . Yes No
Note: *Checking "Yes" will not increase your tax or reduce your refund.*

Requested by Census Bureau for Revenue Sharing ▶
A Where do you live (actual location of residence)? (See page 2 of Instructions.) State | City, village, borough, etc.
B Do you live within the legal limits of a city, village, etc.? ☐ Yes ☐ No
C In what county do you live?
D In what township do you live?

Filing Status

Check only one box.

For IRS use only

1 ☐ Single
2 ☐ Married filing joint return (even if only one had income)
3 ☐ Married filing separate return. Enter spouse's social security no. above and full name here ▶
4 ☐ Head of household. (See page 6 of Instructions.) If qualifying person is your unmarried child, enter child's name ▶
5 ☐ Qualifying widow(er) with dependent child (Year spouse died ▶ 19). (See page 6 of Instructions.)

Exemptions

Always check the box labeled Yourself. Check other boxes if they apply.

6a ☐ Yourself ☐ 65 or over ☐ Blind
b ☐ Spouse ☐ 65 or over ☐ Blind
Enter number of boxes checked on 6a and b ▶

c First names of your dependent children who lived with you ▶
Enter number of children listed on 6c ▶

d Other dependents:

(1) Name	(2) Relationship	(3) Number of months lived in your home	(4) Did dependent have income of $1,000 or more?	(5) Did you provide more than one-half of dependent's support?

Enter number of other dependents ▶

7 Total number of exemptions claimed . . . . . . Add numbers entered in boxes above ▶

Income

Please attach Copy B of your Forms W-2 here.

If you do not have a W-2, see page 5 of Instructions.

Please attach check or money order here.

8	Wages, salaries, tips, etc.	8	
9	Interest income *(attach Schedule B if over $400)*	9	
10a	Dividends *(attach Schedule B if over $400)* ______, 10b Exclusion ______		
c	Subtract line 10b from line 10a	10c	
11	Refunds of State and local income taxes *(do not enter an amount unless you deducted those taxes in an earlier year—see page 9 of Instructions)*	11	
12	Alimony received	12	
13	Business income or (loss) *(attach Schedule C)*	13	
14	Capital gain or (loss) *(attach Schedule D)*	14	
15	40% of capital gain distributions not reported on line 14 (See page 9 of Instructions)	15	
16	Supplemental gains or (losses) *(attach Form 4797)*	16	
17	Fully taxable pensions and annuities not reported on line 18	17	
18	Pensions, annuities, rents, royalties, partnerships, etc. *(attach Schedule E)*	18	
19	Farm income or (loss) *(attach Schedule F)*	19	
20a	Unemployment compensation (insurance). Total received ______		
b	Taxable amount, if any, from worksheet on page 10 of Instructions	20b	
21	Other income (state nature and source—see page 10 of Instructions) ▶	21	
22	**Total income.** Add amounts in column for lines 8 through 21 ▶	22	

Adjustments to Income

(See Instructions on page 10)

23	Moving expense *(attach Form 3903 or 3903F)*	23	
24	Employee business expenses *(attach Form 2106)*	24	
25	Payments to an IRA (enter code from page 10 ....)	25	
26	Payments to a Keogh *(H.R. 10)* retirement plan	26	
27	Interest penalty on early withdrawal of savings	27	
28	Alimony paid	28	
29	Disability income exclusion *(attach Form 2440)*	29	
30	**Total adjustments.** Add lines 23 through 29 ▶	30	

Adjusted Gross Income

31 **Adjusted gross income.** Subtract line 30 from line 22. *If this line is less than $10,000, see "Earned Income Credit" (line 57) on pages 13 and 14 of Instructions. If you want IRS to figure your tax, see page 3 of Instructions* . . . ▶ 31

☆ U.S. GOVERNMENT PRINTING OFFICE: 1980—0-313-250 13-2687299 Form **1040** (1980)

Figure 13-25. IRS Form 1040.

Tax Computation
(See Instructions on page 11)

32	Amount from line 31 *(adjusted gross income)*	32
33	If you do not itemize deductions, enter zero. If you itemize, complete Schedule A (Form 1040) and enter the amount from Schedule A, line 41	33
	Caution: If you have unearned income and can be claimed as a dependent on your parent's return, check here ▶ ☐ and see page 11 of the Instructions. Also see page 11 of the Instructions if: • *You are married filing a separate return and your spouse itemizes deductions, OR* • *You file Form 4563, OR* • *You are a dual-status alien.*	
34	Subtract line 33 from line 32. Use the amount on line 34 to find your tax from the Tax Tables, or to figure your tax on Schedule TC, Part I Use Schedule TC, Part I, and the Tax Rate Schedules ONLY if: • *Line 34 is more than $20,000 ($40,000 if you checked Filing Status Box 2 or 5), OR* • *You have more exemptions than are shown in the Tax Table for your filing status, OR* • *You use Schedule G or Form 4726 to figure your tax.* Otherwise, you MUST use the Tax Tables to find your tax.	34
35	Tax. Enter tax here and check if from ☐ Tax Tables or ☐ Schedule TC	35
36	Additional taxes. (See page 12 of Instructions.) Enter here and check if from ☐ Form 4970, ☐ Form 4972, ☐ Form 5544, ☐ Form 5405, or ☐ Section 72(m)(5) penalty tax	36
37	**Total.** Add lines 35 and 36 ▶	37

Credits
(See Instructions on page 12)

38	Credit for contributions to candidates for public office	38
39	Credit for the elderly *(attach Schedules R&RP)*	39
40	Credit for child and dependent care expenses (*attach Form 2441*)	40
41	Investment credit *(attach Form 3468)*	41
42	Foreign tax credit *(attach Form 1116)*	42
43	Work incentive (WIN) credit *(attach Form 4874)*	43
44	Jobs credit *(attach Form 5884)*	44
45	Residential energy credits *(attach Form 5695)*	45
46	**Total credits.** Add lines 38 through 45	46
47	**Balance.** Subtract line 46 from line 37 and enter difference (but not less than zero) ▶	47

Other Taxes
(Including Advance EIC Payments)

48	Self-employment tax *(attach Schedule SE)*	48
49a	Minimum tax. Attach Form 4625 and check here ▶ ☐	49a
49b	Alternative minimum tax. Attach Form 6251 and check here ▶ ☐	49b
50	Tax from recomputing prior-year investment credit *(attach Form 4255)*	50
51a	Social security (FICA) tax on tip income not reported to employer *(attach Form 4137)*	51a
51b	Uncollected employee FICA and RRTA tax on tips *(from Form W–2)*	51b
52	Tax on an IRA *(attach Form 5329)*	52
53	Advance earned income credit (EIC) payments received (from Form W–2)	53
54	**Balance.** Add lines 47 through 53 ▶	54

Payments
Attach Forms W-2, W-2G, and W-2P to front.

55	Total Federal income tax withheld	55
56	1980 estimated tax payments and amount applied from 1979 return	56
57	Earned income credit. If line 32 is under $10,000, see pages 13 and 14 of Instructions	57
58	Amount paid with Form 4868	58
59	Excess FICA and RRTA tax withheld (two or more employers)	59
60	Credit for Federal tax on special fuels and oils *(attach Form 4136 or 4136–T)*	60
61	Regulated Investment Company credit *(attach Form 2439)*	61
62	**Total.** Add lines 55 through 61 ▶	62

Refund or Balance Due

63	If line 62 is larger than line 54, enter amount **OVERPAID** ▶	63
64	Amount of line 63 to be **REFUNDED TO YOU** ▶	64
65	Amount of line 63 to be applied to your 1981 estimated tax ▶	65
66	If line 54 is larger than line 62, enter **BALANCE DUE.** Attach check or money order for full amount payable to "Internal Revenue Service." Write your social security number on check or money order ▶ (Check ▶ ☐ if Form 2210 (2210F) is attached. See page 15 of Instructions.) ▶ $	66

Please Sign Here

Under penalties of perjury, I declare that I have examined this return, including accompanying schedules and statements, and to the best of my knowledge and belief, it is true, correct, and complete. Declaration of preparer (other than taxpayer) is based on all information of which preparer has any knowledge.

▶ Your signature Date | ▶ Spouse's signature (if filing jointly, BOTH must sign even if only one had income)

Paid Preparer's Use Only

Preparer's signature and date ▶		Check if self-employed ▶ ☐	Preparer's social security no.
Firm's name (or yours, if self-employed) and address ▶		E.I. No. ▶	
		ZIP code ▶	

Figure 13-25 continued.

33. Prepare a template as in the previous problem; however, prepare it for the Form 1040 shown in Figure 13-25.

34. Prepare a spreadsheet to help decide on a source from which to purchase a microcomputer. Prepare a row for each item to be purchased, for example, microcomputer, disk drives, monitor, software, etc. Then prepare a column with the manufacturer's suggested retail price. For each source prepare three columns: the quoted price, the dollar discount from the suggested retail price (if any), and the percentage discount. Total columns as appropriate.

35. To gain practice in entering labels, prepare a template that you can use as the format of a weekly calendar. In one column include the times of day broken into convenient units (for example 8:00, 8:15, 8:30, etc.). Then include columns titled Monday, Tuesday,...Sunday. Include an area to enter the dates for the week.

36. Prepare a template in which the top row contains the column labels (A, B, C, D...) and the first column contains the row labels (1, 2, 3, 4...). This template itself can be useful as a tool on spreadsheets from which printed output will be prepared. Printing these labels with the report can be useful in identifying specific entries of the spreadsheet that may be changed later. Figure 13-26 shows part of the template that should be produced for this problem. For practice, prepare two spreadsheets, one with left justified labels and one with right justified labels.

When this worksheet is complete it can be used as an overlay on another sheet we want printed. For example, with the other sheet in memory we can move the cursor to location A1, and insert both a row and a column. Then load the template of this problem. Then print. The row and column labels will be included in the printed report.

Remember later, when working on the sheet, to delete these labels since they do cause all coordinate references to be revised.

37. As a student, suppose that you want to record a running average of your grade for the course. Prepare a template with columns for the assignment or test number (for example, assignment 1, test 4, etc.), your score, the possible score, your total to date, the total possible to date, and your average. An example is shown in Figure 13-27.

38. An animal hospital provides boarding services for cats ($4 per day), small dogs ($5 per day) and large dogs ($6 per day). Prepare a table with four columns, one column for the length of

```
A1  B   C   D   E   F   G   H   I   J   K   L   M   N   O   P
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
```

Figure 13-26. A template in which the row and column labels are included in the spreadsheet itself. Such a template could serve as an overlay on other templates.

the stay in days, and then one column for each type of animal that lists the total cost of a stay at the hospital. Print the table for lengths of stay from 1 to 60 days.

NAME ______________		CLASSROOM GRADING SHEET			
DESCRIPTION	MY SCORE	POSSIBLE	TOTAL TO DATE	TOTAL POSSIBLE	%
ASSIGN. 1-1	______	20	______	20	____
ASSIGN. 1-2	______	20	______	40	____
ASSIGN. 1-3	______	25	______	65	____
ASSIGN. 1-4	______	25	______	90	____
ASSIGN. 1-5	______	60	______	150	____
TEST 1	______	100	______	250	____
ASSIGN. 1-6	______	25	______	275	____
ASSIGN. 1-7	______	70	______	345	____
ASSIGN. 1-8	______	40	______	385	____
TEST 2	______	100	______	485	____
ASSIGN. 1-9	______	50	______	535	____
ASSIGN. 1-10	______	25	______	560	____
TEST 3	______	100	______	660	____
ASSIGN. 1-11	______	90	______	750	____
ASSIGN. 1-12	______	50	______	800	____
FINAL EXAM	______	200	______	1000	____

Figure 13-27. A sample recordkeeping sheet for individual grades.

39. A gas and electric company charges rates as below:

		Rate
First	3 therms, or less	$2.90000
Next	997 therms, per therm	.43709
Next	99,000 therms, per therm	.37672
Over	100,000 therms, per therm	.34191

The minimum monthly charge is $2.90.

Prepare a template on which the number of therms consumed is entered and that computes the total charge. (A review of the @MAX and @MIN functions may be helpful.)

40. Prepare a report like the one in Figure 13-28 which shows projected space requirements for several departments of a data processing division. The column, Staff Total, is entered by the user. The Type-of-Office column is a code which is used to look up an office size in square feet (SF) which is then placed in the Unit-SF column. The codes and their associated areas are

Type of Office	Unit SF
1	160
2	140
3	120
4	100
5	80

Using a table allows us to vary office size by changing these values in the table and then immediately determining the impact on the total space requirements.

DATA PROCESSING DIVISION SPACE REQUIREMENTS

Code No.	Space Title	Staff Total	Type of Office	Unit SF	Total Square Feet
448	ADMIN.				
	- Division Director	1	1	160	160
	- Secretary	5	5	80	400
		6		Subtotal	560
442	SYSTEMS				
	- Manager	1	3	120	120
	- Offices	10	5	80	800
	- Users' Area			1,300	1,300
	- Gen'l Area			2,500	2,500
	- Storage			160	160
		11		Subtotal	4,880
442	PROGRAMMING				
	- Manager	1	3	120	120
	- Offices	21	5	80	1,680
	- Terminal Area			1,000	1,000
	- Gen'l Area			3,000	3,000
		22		Subtotal	5,800
448	OPERATIONS				
	- Manager	1	3	120	120
	- Offices	7	5	80	560
		8		Subtotal	680
450	OTHER				
	- Reception Area			260	260
	- Resource Area/Supply Area			80	80
	- Terminal Room			80	80
	- Storage			160	160
	- Mail Box Area			80	80
	- Coffee/Break Area			80	80
				Subtotal	740
				NET TOTAL	12,660

Figure 13-28. A space requirements planning sheet.

VisiCalc Summary Reference

COMMANDS

Symbol	Name	Purpose
/B	the Blank command	Blank a single entry.
/C	the Clear command	Clear the entire sheet, Y confirms.
/D	the Delete command	Delete a row or column.
	R	Delete a row.
	C	Delete a column.
/F	the Format command	Format a single entry.
	D	Default
	G	General
	I	Integer
	L	Left justified
	R	Right justified
	$	Dollars and cents
	*	Graph
/G	the Global command	Globally affect the sheet.
	C	Column width
	O	Order of recalculation
	R	Rowwise
	C	Columnwise
	R	Recalculation
	A	Automatic
	M	Manual
	F	Format
	D	Default
	G	General
	I	Integer
	L	Left justified
	R	Right justified
	$	Dollars and cents
	*	Graph
/I	the Insert command	Insert a row or column.
	R	Insert a row
	C	Insert a column
/M	the Move command	Move a row or column.

/P	the Print command	Print all or part of the sheet to a device.
/R	the Replicate command	Replicate an entry, row, or column onto another area of the sheet.
/S	the Storage command	Access storage.
	L	Load a sheet
	S	Store a sheet
	D	Delete a stored sheet
	I	Initialize a diskette
	Q	Quit
	#	Access a file in the Data Interchange Format (DIF, a trademark of Software Arts, Inc.)
	L	Load a DIF file
	S	Save a DIF file
/T	the Title command	Freeze titles in place.
	H	Horizontally
	V	Vertically
	B	Both
	N	Neither
/V	the Version Number command	Display the version number.
/W	the Window command	Affect the window display.
	H	Horizontal split
	V	Vertical split
	1	1 (one) window
	S	Synchronized scrolling
	U	Unsynchronized scrolling
/-	the Repeating Label command	Repeat part of a label throughout a single entry.

BUILT-IN FUNCTIONS

@ABS(argument)	the absolute value
@ACOS(argument)	the arccosine
@ASIN(argument)	the arcsine
@ATAN(argument)	the arctangent
@AVERAGE(argument1,argument2,...)	the average
@COS(argument)	the cosine
@COUNT(argument1,argument2...)	count how many
@ERROR	the error function and error value
@EXP(argument)	*e* to a power
@INT(argument)	integer
@LN(argument)	natural logarithm
@LOG10(argument)	logarithm, base 10
@LOOKUP(argument1,argument2)	look up a value in a table
@MAX(argument1,argument2,...)	the maximum value
@MIN(argument1,argument2,...)	the minimum value
@NA	not available
@NPV(argument1,argument2)	the net present value
@PI	value of π
@SIN(argument)	the sine
@SQRT(argument)	the square root
@SUM(argument1,argument2,...)	sum the values
@TAN(argument)	the tangent

Bibliography Of VisiCalc Articles

"Administrators in Vocational Training Save Time with Apples." *Apple Education News* 2, no. 3 (November 1981): 5. VisiCalc use in projecting teachers' salaries when bargaining.

Ahl, David H. "Evaluation of VisiTrend and VisiPlot from Personal Software." *Creative Computing* 7, no. 12 (December 1981): 98, 100, 102, 104; and a related letter to the editor, "GRM of an Idea" from Reed Jenney 8, no. 3 (March, 1982): 8. Discussion of these products, with an aside about frustrations in using VisiCalc, and product interfaces with DIF, a data interchange.

Alsop II, Stewart. "How VisiCalc Works." *INC.* 3, no. 11 (November 1981): 104-105. Introductory description of VisiCalc.

Alsop II, Stewart. "Software Arts Wrote the First Best-Seller." *INC.* 4, no. 1 (January 1982): 71, 73, 75. Interesting article on the development of VisiCalc by Dan Bricklin and Bob Frankston and its marketing by Dan Fylstra. Includes mention of competing products.

Anbarlian, Harry. *An Introduction to VisiCalc Matrixing with Apple and IBM.* New York, NY: McGraw-Hill, 1982, 252 pages.

Anders, George. "Experts Say IBM's Entry Will Buoy Already Booming Software Industry." *The Wall Street Journal,* Wednesday, August 19, 1981, 25. Includes reference to VisiCalc availability on the IBM Personal Computer.

"Apple-Based Visi-Schedule Joins VisiCorp Line." *Computer Business News* 5, no. 7 (February 15, 1982): 16. Relationship of this product to VisiCalc is presented.

"At 3.3, VisiCalc Spawns a Family." *Softalk* 1, no. 10 (June 1981): 31-32. Describes related 'Visi' products and their interrelationship through DIF, a data interchange.

Barry, Tim. "SuperCalc Spread-Sheet Simulator from Sorcim Corp." *InfoWorld* 3, no. 20 (October 5, 1981): 30-32. Discusses absence of VisiCalc from the CP/M operating system, describes SuperCalc, and compares the two.

Batt, Robert. "Firm Tries Personal Computer Transformation, Unique Approach." *Computerworld* 16, no. 19 (May 10, 1982): 115. Describes the MBA product from Context Management Systems, Inc. including command language similarities to VisiCalc.

Batt, Robert. "Micro Boom Hits Corporate Managers' Desks." *Computerworld* 16, no. 7 (February 22, 1982): 4. Use of microcomputers in a stand-alone mode, including VisiCalc use.

Bayer, Barry D. "Visulating." *Desktop Computing* (November 1981): 84-88. The first of a periodic column on VisiCalc; and a related letter to the editor from Joshua B. Malks, 2, no. 5 (May 1982): 97, praising the column.

Berger, Myron. "Scenarios For Success: The Vision of Spreadsheeting." *Personal Computing* 6, no. 4 (April 1982): 58-59, 61-63. General discussion of spreadsheet programs (with listings of 22 programs), several specific business applications, and of VisiGroup (see *SPREADSHEET* below).

Blakeney, Susan. "Applefest Attracts DPers from Across the Country." *Computerworld* 16, no. 21 (May 24, 1982): 20-21. Contains brief quotes from an attendee on VisiCalc 'what if' value available on micros rather than mainframes.

Blank, George. "Break Even Analysis with VisiCalc." *Creative Computing* 7, no. 5 (May 1981): 96-98. Short general discussion of VisiCalc, with a detailed break even template.

Bricklin, Dan, and Bob Frankston. *VISICALC Computer Software Program for the Apple II and II Plus.* Software Arts, Inc., Sunnyvale, CA: Personal Software Inc., 1979. The product manual.

"Bricklin Wins Hopper Award." *Computerworld* 15, no. 37 (September 14, 1981): 30. Association for Computing Machinery awards Dan Bricklin the Grace Murray Hopper Award in recognition of computing achievement made by a person before his 30th birthday.

Bride, Edward J. "Hardware Vendors See Software as the Difference." *Software News* (May 3, 1982): 37. Reports Dan Fylstra's (VisiCorp) comments on the advantages of hardware/software vendors over software-only vendors.

Bunnell, David. "The Man Behind the Machine." *PC* 1, no. 1 (February-March 1982): 16-23. An interview with Bill Gates of Microsoft including discussion of VisiCalc and Multiplan.

"Business Software Update: Financial Planning, Text Processing and Accounting." *Personal Computing* 6, no. 5 (May 1982): 190, 192. Describes the GRAPH product from Software Publishing Corp., and its ability to use VisiCalc files.

Carroll, Nanette Rundel. "Leading Executives to Computers." *Data Training* 1, no. 3 (February 1982): 10-12. Describes training of executives of the Bank of America in data processing concepts, including demonstration of VisiCalc.

Castlewitz, David M. and Lawrence J. Chisausky, with Patricia Kronberg. *VisiCalc Home and Office Companion.* Berkeley, CA: OSBORNE/McGraw-Hill, 1982, 181 pages. Presents 50 VisiCalc models for home and business use.

"Computer Stores: Tantalizing Opportunity Selling Computers to Customers." *Business Week* no. 2707 (September 28, 1981): 76-80, 81-82. Includes brief comments of Dan Fylstra of VisiCorp, Personal Software, which markets VisiCalc.

"Computer Takes Drudgery Out of Record Keeping." *Farm Computer News* 1, no. 6 (October 1981): 1-2. VisiCalc usage in maintaining farm records.

"Computer Using Educators (CUE) Position Paper on Commercial Software Pricing Policies." *The Computing Teacher* 8, no. 6 (academic year 1980-81): 53-54. Briefly discusses the negative effects of the VisiCalc pricing policy on its otherwise high potential as a teaching tool.

Couch, John. "Datagramming." *Apple Orchard* 2, no. 4 (Winter 1981/82): 14-16, 18-19. Includes brief discussion of VisiCalc as a tool allowing non-programmers to use a computer, but having fixed input and output and no sorting or query capabilities.

Cowen, Robert. "Housebreaking the Software." *Technology Review* 85, no. 1 (January 1982): 6, 21. Cites the success of VisiCorp, Personal Software and Software Arts with VisiCalc.

Cox, Terry. "Personal Computers Leave the Nest." *Computerworld* 16, no. 18 (May 3, 1982): In Depth 1-4, 6. Brief mention of VisiCalc credited with sales of Apple and TRS-80 computers.

Craig, John. "VisiCalc From Software Arts." *InfoWorld* 2, no. 9 (June 9, 1980): 14, 19, 27. Software 'report card' giving VisiCalc excellent marks.

"Datapro's User Ratings of Proprietary Software." Reported in *Computerworld* 15, no. 51 (December 21, 1981): 17, 26. VisiCalc was one of 19 programs rated highest from a total of 417 evaluated by users.

de Peyster, Deborah. "Company Building Software Pipeline." *Computer Business News* 5, no. 20 (May 17, 1982): 1, 8. Announces VisiCalc on DEC microcomputers.

de Peyster, Deborah. "Creator of VisiCalc Won't Sign This One Away." *Computer Business News* 5, no. 21 (May 24, 1982): 5. Announces Software Arts' new product, TK!Solver, and their intention to market it themselves.

"DEC Signs Software Suppliers." *Computerworld* 16, no. 2 (May 17, 1982): 4. Brief article on software, including VisiCalc, to be available on DEC's personal computers.

"Dedication to Education." *The Individualized Learning Letter (TILL) Micro-Ed Digest* 11, no. 2 (November 1981): 1, 5, 6. Comments from a speech by Glen Polin, Apple Computer, Inc., crediting VisiCalc with selling more microcomputers than any other piece of software.

"Digicalc More Than a Look-Alike, Claims Vendor." *Computer Business News* 5, no. 2 (January 11, 1982): 21. Digicalc, for the DEC marketplace, is described, including a brief comparison to VisiCalc by the marketing director of Why Systems Inc., its vendor.

Ditlea, Steve. "The New Software: A Powerful Ally." *INC* 3, no. 11 (November 1981): 103-105, 108. Discusses a variety of business software, beginning with VisiCalc.

Ditlea, Steve. "Second-Generation Word-Processing Programs." *Popular Computer* 1, no. 8 (June 1982): 38, 42, 44, 46, 48. Discussion of word processing packages, including the Executive Secretary and Letter Perfect and their ability to access VisiCalc files, as well as mention of the codes of Apple Writer 2.0 in relation to VisiCalc.

Ditlea, Steve, and Joanne Tangorra. "The Birth of an Industry." *INC.* 4, no. 1 (January 1982): 64-66, 68, 70. The origins and growth of a number of software companies and the individuals who founded them, including Software Arts and VisiCorp, Personal Software.

Droegemueller, Lee. "Enrollment Reporting Using VisiCalc." *TRS-80 Microcomputer News* 3, no. 7 (March 1981).

Edlin, Jim. "EasyWriter To Get Improvements." *PC* 1, no. 2 (April-May 1982): 114. Brief description of using the Easywriter word processing program on the IBM Personal Computer to work with VisiCalc files.

Fastie, Will. "IBM Images." *Creative Computing* 8, no. 1 (January 1982): 164, 166, 168, 170-171. Introduction to the IBM Personal Computer, including mention of VisiCalc availability.

Fastie, Will. "IBM Images." *Creative Computing* 8, no. 6 (June 1982): 206, 208, 210, 212-214. Includes a discussion of memory expansion boards for the IBM Personal Computer which can be used with VisiCalc.

Fenstermacher, Sandi, and Susan Foster. "An Easement for Frustration - Automating the Real Estate Office." *Personal Computing* 5, no. 8 (October 1981): 101, 102, 104-105.

Ferrarini, Elizabeth M. "Invasion of the Microcomputers." *Computerworld* 15, no. 22 (June 1, 1981): In/Depth 12-22. Includes discussion of VisiCalc usage at several firms.

Fields, Randy. "A Financial Budgeting Model with Comments." *Apple Orchard* 3, no. 1 (March-April, 1982): 58-64. Presentation of a VisiCalc template for a budgeting system for users' groups of the Apple ///.

"Financial Modeling." *Datamation*, April 1981: 244. A brief description of FPL (Financial Planning Language) with a reference to VisiCalc.

Foster, Susan. "Consciousness Raising by Computer." *Personal Computing* 5, no. 6 (August 1981): 20-24, 88, 91. Discussion of computing changing the way individuals think, including the use of VisiCalc.

Fox, Tom. "Hewlett-Packard HP 125: System of the Month." *Interface Age* 6, no. 12 (December, 1981): 64, 66-67. Includes discussion of VisiCalc/125 for the HP 125.

Fox, Tom. "Hewlett-Packard HP-83A: System of the Month." *Interface Age* 6, no. 5 (May 1981): 42-48. Includes discussion of VisiCalc Plus for the HP-83A.

Fox, Tom. "Personal Software's VisiCalc: Software Product Reviews." *Interface Age* 5, no. 5 (May 1980): 144. VisiCalc reviewed, including very positive comments on the product.

Frank, Werner L. "The Evolution of Microcomputer Software." *Computerworld* 16, no. 17 (April 26, 1982): 41-43. VisiCalc helps move the focus of microcomputers "from games to business."

Freiberger, Paul. "Atari Private Detective Nabs Software Pirate." *InfoWorld* 3, no. 32 (December 28, 1981): 1, 3. Discussion of copying VisiCalc and software piracy.

"From Bessie to 'Number 365.'" *Personal Computing* 6, no. 2 (February 1982): 16. Use of VisiCalc to do "the mathematics" of cattle maintenance.

"The Future of Personal Computers." Boston Computer Society's second annual Forum on the Future of Personal Computers, October 15, 1981. Audio tapes. Morris Plains, NJ: *Creative Computing*.

Fylstra, Dan, and Bill Kling. *VisiCalc ATARI 800 32K*. Sunnyvale, CA: Personal Software, Inc., 1979. The product manual.

Fylstra, Dan, and Bill Kling. *VisiCalc User's Guide for the Apple II and II Plus, 16 Sector Version*. Sunnyvale, CA: Personal Software Inc., 1981. The product manual.

Gabel, David. "A Personal Computing Interview with Bob Harp." *Personal Computing* 6, no. 3 (March 1982): 19-20, 108, 110, 113-115. Brief discussion of VisiCalc and ExecuPlan.

Gibney, Frank Jr. "The Tail that Wags the Dog." *Newsweek* February 22, 1982, 55. Brief summary of the success of Dan Bricklin and Bob Frankston with VisiCalc, quoting from the Electronics Letter of Ben Rosen.

Gold, Jordan. "Keep Your Costs Bite-Size." *Personal Computing* 6, no. 2 (February 1982): 18-20, 22, 24, 153-154. Cost controlling in a variety of businesses with VisiCalc and other software; includes cash flow and "what if . . ." computations in tax analysis.

Good, Phillip. "Beyond VisiCalc." *Popular Computing* 1, no. 3 (January 1982): 38, 40-42. Descriptions of various business software products.

Good, Phillip. "VisiCalc, An Electronic Worksheet." *Popular Computing* 1, no. 2 (December 1981): 34, 36, 38. General product description with references to several specific uses and comparisons to SuperCalc.

Gordon, Robert. "Q & A: Portia on Personal Computers." *Computer Career News* 2, no. 6 (April 20, 1981): 4-5, 11. An interview with Portia Isaacson in which VisiCalc is discussed.

Green, Doug. "VisiCalc: Reason Enough for Owning A Computer." *Creative Computing* 6, no. 8 (August 1980): 26-28. Product evaluation and capability summary.

Green, Douglas and Denise Green. "VisiCalc Templates, Real Estate Analyst Series." *InfoWorld* 4, no. 18 (May 10, 1982): 44-45, 47. A software review of this product.

Greitzer, John. "More Memory, IBM Lure Corporate Giants." *Com-*

puter Business News 5, no. 17 (April 26, 1982): 17. Mention of VisiCalc and AgriCalc uses in agriculture.

Heck, Mike. "The Commodore Logbook: VisiCalc Applications." *Interface Age* 7, no. 4 (April 1982): 48-51, 150, 152, 153. General description of VisiCalc with an example from an advertising agency.

Heintz, Carl. "Business Software Review: VisiCalc Examined." *Interface Age* 6, no. 3 (March 1981): 46-47. General product description.

Heintz, Carl. "Major Additions to VisiCalc Revealed." *Interface Age* 6, no. 8 (August 1981): 70-72, 148-149. Description of VisiCalc and related products from VisiCorp, Personal Software, including product interfacing with DIF, a data interchange.

Hewes, Jeremy Joan. "A Tale of Two Beginnings." *PC* 1, no. 2 (April-May 1982): 89, 91, 95-96. Includes VisiCalc usage on an IBM Personal Computer for financial analysis and forecasting in the treasurer's department of a drug wholesaler.

Hewes, Jeremy Joan, "IBM's New Concept, Taking the Measure, Part II: A Glimpse at Two PC Manuals." *PC* 1, no. 2 (April-May 1982): 116-118. Discussion of the VisiCalc manual for the IBM Personal Computer, including comparison with the Apple version.

Hewin, Larry M. and Duane Saylor. "Issues and Answers." *Desktop Computing* 2, no. 6 (June 1982): 8, 10-11. Includes general description of spreadsheets and VisiCalc.

Hixson, Amanda. "VisiTrend and VisiPlot, Trend Analysis and Plotting." *InfoWorld* 4, no. 17 (May 3, 1982): 34-36. Description of this product including interfacing with VisiCalc.

Hogan, Thom. "Thom Hogan Picks Top Ten." *InfoWorld* 3, no. 5 (March 16, 1981): 4-5, 39. VisiCalc rates second (after CP/M) on this list of the Top Ten microcomputer products.

Hoxie, Gib, and Paul Kircher. "The Model Manager." *Desktop Computing* 1, no. 2 (November 1981): 14-16. Brief mention of VisiCalc as a management tool.

Huntley, David E. "The Other End of the Spectrum." *Softalk* 2, no. 8 (April 1982): 10. Letter to the editor discussing capabilities of The Universal Business Machine that are similar to VisiCalc with negative comments regarding the length of time necessary to learn VisiCalc.

Judd, Dorothy H. "Administrative Decision Tools: A Microcomputer Reality Available Now." *Educational Computer* 1, no. 2

(July/August 1981): 6-7. Includes reference to the use of VisiCalc in solving a variety of problems in educational administration.

Judd, Dorothy H. "Learning to Program Really Isn't Necessary, or BASIC Isn't Basic Anymore." *Educational Computer* 1, no. 3 (September-October 1981): 10-11.

Kalish, Candace E. and Malinda F. Mayer. "DIF: A Format for Data Exchange Between Applications Programs." *BYTE* 6, no. 11 (November 1981): 174, 176, 178, 180, 182, 186, 188, 190, 192, 194, 196, 198, 200, 202, 204, 206.

Kehoe, Louise. "Engineers Get Personal with Computers." *Personal Computing* 5, no.8 (October 1981): 24-26, 29, 31-32.

Kleper, Mike. "Softcover Evolving into Software." *DIPpy*. (Direct Input Phototypesetting Newsletter) 3, no. 10 (December 1981): 1. Briefly discusses the 'publishing' of software, including VisiCalc.

Knudsen, Richard. "A Self-Taught Programmer Insures His Future." *Softalk* 1 (April 1981): 46-48. A discussion of an insurance salesman/broker's experience in programming, including the use of VisiCalc.

Latamore, Bert. "Monroe Sells Solutions." *Desktop Computing* 2, no. 6 (June 1982): 12-17. Describes the Monroe 8820 microcomputer and its use of SuperCalc with brief comparison to VisiCalc.

Light, Christopher U. "Ventures with VisiCalc." *Softalk* 2, no. 5 (January 1982): 158, 160, 162, 168; and a related letter to the editor from Paul F. Swinford, 2, no. 7 (March 1982): 10. (*Refer also* to the Frank Malone, Joe Shelton, Craig Stinson, and Otto K. Wetzel entries in this bibliography.)

Malone, Frank. "Ventures with VisiCalc." *Softalk* 1 (March 1981): 35. (*Refer also* to the Christopher U. Light, Joe Shelton, Craig Stinson, and Otto K. Wetzel entries in this bibliography.)

Maloney, Eric (compiler). "MICRO-SCOPE: Down on the Farm." *Kilobaud Microcomputing* 5, no. 8 (August 1981): 30-31. Computer use on the farm, including brief mention of VisiCalc.

Marbach, William D., Gerald C. Lubenow, William J. Cook, Frank Gibney Jr., and Kim Willenson. "To Each His Own Computer." *Newsweek* February 22, 1982, 50-56. Discussion of a variety of computers, including the role of VisiCalc in the success of Apple Computer, Inc.

Markoff, John. "Computer-Rental Firm Mixes Business and Fun." *InfoWorld* 4, no. 5 (February 8, 1982): 19. Describes a retail business in the computer field which teaches VisiCalc classes among other activities.

McClellan, Jim. Letter to the editor regarding VisiCalc use in teacher salary projections. *The Computing Teacher* 9, no. 8 (April 1982): 5.

McLamb, Ken. "Easier Economic Modeling with Personal Computers." *Personal Computers* 5, no. 12 (December 1981): 31-32, 34, 36, 39. General modeling discussion including a specific example of VisiCalc use by the Mexican government for economic forecasting.

McLamb, Ken. "Personal Computers in Business - The View from the Data-Processing Shop." *Personal Computing* 5, no. 7 (September 1981): 30-31, 35-36, 38, 40.

McLamb, Ken. "Persuasion Made Easy with Powerful Presentations." *Personal Computing* 6, no. 2 (February 1982): 34-37, 39, 84. Using graphics packages, including VisiTrend/Plot and its relationship to VisiCalc.

Meserve, Everett T. (Bill). "The Future of Rabbits." *Datamation* 28, no. 1 (January 1982): 130-131, 133, 136. Brief mention of VisiCalc as an 'easy-to-use' software package for managers amid general discussion of the personal computer industry.

Micro-Calc Business Users' Group Newsletter. 1, no. 1 (May 1981). (P.O. Box 12039, Salem, Oregon 97309)

"Micro Software Industry Seen Confusing." *Computerworld* 16, no. 16 (April 19, 1982): 78. Summarizes comments of Andy Johnson-Laird that products like VisiCalc will survive as long as it costs less to use them than to change to other products that are better engineered.

Milich, Melissa. "The Apple in Big Business." *Softalk* 2, no. 1 (September 1981): 58-59, 61-62. Includes description of use of VisiCalc at Avery International.

Miller, Jonathan. "Lutheran Social Servants Swear By Zardax." *Softalk* 2, no. 9 (May 1982): 80-81, 83, 85. A description of a social service agency's use of an Apple, including VisiCalc as a budgeting tool.

Morgan, Chris. "Can We Agree on Standards?" *BYTE* 6, no. 11 (November 1981): 6-7.

Morgan, Chris. "Japan and the '64K' Question." *BYTE* 7, no. 5 (May 1982): 6, 8, 10. Mention of Japanese computer manufacturers obtaining licenses to use products such as VisiCalc.

Morgan, Chris. "The 'Visiclones' Are Coming." *BYTE* 7, no. 2 (February 1982): 6, 8. Brief mention of several competing spreadsheets as imitations of VisiCalc.

Murphy, Thomas P. "Look for the Kid with the Screwdriver." *Forbes* 129, no. 4 (February 15, 1982): 118. Brief references to VisiCalc among comments of a new user of personal computers.

Needle, David. "Micros Help Athletes With Investments & Negotiations." *InfoWorld* 3, no. 32 (December 28, 1981): 19. Details personal computer and VisiCalc use in the area of consulting for professional athletes.

Needle, David. "Micro Software Helps Users Prepare for the Taxman." *InfoWorld* 4, no. 6 (February 15, 1982): 12-13. Discussion of tax related software including VisiCalc as a general purpose tool available for tax planning.

Nelson, Ted. "Interactive Systems and the Design of Virtuality." *Creative Computing* 6, no. 11 part 1, (November 1980): 52-62; 6, part 2, no. 12 (December 1980): 95-106. Discussion of the design of interactive systems, and how they "seem" to the user; this includes comments on VisiCalc.

"New Business Software Packages." *COMPUTE!* 3, no. 6 (June, 1981): 147-148. Describes VisiCorp, Personal Software, products, including VisiCalc.

O'Connor, Rory J. "Apple Denies Computer Will Make NCC Debut." Speculates on Apple Computer, Inc. 'Lisa' computer and on VisiCalc's availability on it.

O'Connor, Rory J. "Conferees Stress Need for 'Users' Software." *Computer Business News* 4, no. 43 (October 26, 1981): 1, 3, 5.

O'Connor, Rory J. "Micros Can File-Swap With DIF Data Format." *Computer Business News* 5, no. 7 (February 15, 1982): 1, 5; and a related letter to the editor from Bill Gurley, 5, no. 15 (April 12, 1982): 8.

O'Connor, Rory J. "PC Applications, OSs, Languages." *Computer Business News* 5, no. 16 (April 19, 1982): 6. Discussion of SuperCalc for the IBM Personal Computer with brief mention of VisiCalc.

O'Connor, Rory J. "PC Packages Just Reaching 'Pipeline'." *Computer Business News* 5, no. 17 (April 26, 1982): 1, 8. Cites success of VisiCalc on the IBM Personal Computer.

O'Connor, Rory J. "Software Sales Rate Called 'Frightening'." *Computer Business News* 5, no. 3 (January 18, 1982): 6. Comments on the high growth rate expected for software, including VisiCalc.

Paul, Lois. "McCormack & Dodge Exec Hints at Support for Micro Software." *Computerworld* 15, no. 44 (November 2, 1981): 14.

Discusses plans for two-way exchange of data between microcomputers and mainframes.

Patchen, Steve. "A Review of VisiCalc & T/Maker." *Lifelines* 1, no. 9 (February 1981): 2-6. Comparison of these two products.

Perry, Robert. "Financial Modeling Software: Tools for the Overworked Manager." *Personal Computing* 5, no. 6 (June 1981): 22-28, 59-70, 108. This lengthy article compares approximately 15 financial modeling software programs and includes a series of questions to ask when considering various packages.

"*Personal Computing* Interview with Ben Rosen." *Personal Computing* 6, no. 6 (June 1982): 28-30, 98, 100-101. Includes brief mention of VisiCalc.

"*Personal Computing* Interview with Jim Dowe." *Personal Computing* 6, no. 5 (May 1982): 30-31, 86, 88, 90. Discusses the time necessary to learn VisiCalc, estimated here at between four and twenty hours.

Pitt, Joel. "DSS/F, a Financial-Modeling and Reporting Package." *InfoWorld* 4, no. 19 (May 17, 1982): 43-46. Discussion of DSS/F by Addison-Wesley, with brief comparisons to VisiCalc.

Pollack, Andrew. "Next, a Computer on Every Desk." *The New York Times*, August 23, 1981, Section 3 (Business): 1, 15. Includes discussion of the rise in popularity of Apple Computer, Inc., based in part on VisiCalc.

Pournelle, Jerry. "SuperCalc, Spelling Programs, BASIC Compilers, and Home-Grown Accounting." BYTE 7, no. 5 (May 1982): 226, 228-230, 232, 234, 236, 238, 240, 242-243. Description of SuperCalc (with brief mention of its relationship to VisiCalc) and a tale of installing it on a Godbout 8085/88 with a Zenith Z-19 terminal.

"Programmer's Guide to the Data Interchange Format." Cambridge, MA: *The DIF Clearinghouse*. A guide to the language for data developed by Software Arts, Inc. (P.O. Box 70, MIT Branch, Cambridge, MA 02139).

"Proprietary Software: User Ratings and Cluster Analysis." Datapro Research Corp., Delran, NJ 08075. Reported in *Computerworld* 15, no. 51 (December 21, 1981): 16-26, 30-33. (*Refer also* to "Datapro's . . . " entry above.)

Radding, Alan. "A Window on the Weather." *Personal Computing* 6, no. 2 (February 1982): 49-52, 90, 92, 94-95. Personal computer use in weather forecasting including brief mention of VisiCalc.

Radding, Alan. "Have Your Cost Estimates Lost Credibility?" *Per-*

sonal Computing 6, no. 4 (April 1982): 30-31, 33. Cost estimating in the commercial building industry using VisiCalc.

Radding, Alan. "Rapid Reading of Research Data." *Personal Computing* 6, no. 1 (January 1982): 42-44, 47, 48, 148. Personal computer use in research including VisiCalc and other software.

Ramsdell, Robert E. "The Flexibility of VisiPlot." *BYTE* 7, no. 2 (February 1982): 32, 34, 36. Review of this product including data transfer from VisiCalc using DIF, a data interchange.

Ramsdell, Robert E. "The Power of VisiCalc: Product Review." *BYTE* 5, no. 11 (November 1980): 190-192. Early review of VisiCalc with strengths and program constraints and discussion of DIF, a data interchange.

"Roaming Hi-Tech Pirates." *Time* 119, no. 6 (February 8, 1982): 61. Brief discussion of the illegal copying of VisiCalc and other software.

Robbins, Paul. "Absent-Minded VisiCalc." *Softalk* 2, no. 8 (April 1982): 19. Letter to the editor describing a problem with the catalog of a diskette of VisiCalc files; and a letter responding to the problem from Sandy Pierce, "Tricky Lifesaver," 2, no. 9 (May 1982): 15, 17 describing software to solve the problem of a damaged catalog on a diskette.

Robins, Clark. "Despite Piracy, the System Works." *InfoWorld* 4, no. 16 (April 26, 1982): 23. Letter to the editor encouraging placement of VisiCalc in public libraries as a means of promoting use of the product with subsequent increases in sales.

Ryan, Paul. "Intelligent Investment Management by Computer." *Personal Computing* 6, no. 1 (January 1982): 38-40, 98-99. Includes VisiCalc use in managing investments.

"Sagas of Five Who Made it: Sizzling Software." *Time* 119, no. 7 (February 15, 1982): 42, 44. Profile of Dan Fylstra of VisiCorp, Personal Software, which markets VisiCalc.

Sanders, Bill. "Enough of Buccaneers and Barons." *Softalk* 2, no. 7 (March 1982): 18, 20. VisiCalc included as an example within a letter to the editor summarizing viewpoints of the 'pirates' and 'manufacturers' of software.

SATN (Software Arts Technical Notes). *The Journal for VisiCalc Users*, Software Arts, Inc., P.O. Box 815, Quincy, MA 02169.

Seaman, John. "Microcomputers Invade the Executive Suite." *Computer Decisions* 13, no. 2 (February 1981): 68-73, 158-159, 165-174. A thorough discussion of the topic, including the use of VisiCalc and other software.

Shapiro, Neil. "Visible Calculator." *Popular Mechanics* 152, no. 4 (October 1979): 49. Early product review of VisiCalc as a product that "will revolutionize personal computing."

Shelton, Joe. "Ventures with VisiCalc." *Softalk* 2, no. 8 (April 1982): 143, 145-146, 148. (*Refer also* to Christopher U. Light, Frank Malone, Craig Stinson, and Otto K. Wetzel entries in this bibliography.)

"The Smash Hit of Software." *Time* 118, no. 9 (March 2, 1981): 69. Short VisiCalc history.

Smolin, Rocky. "CalcStar: Good Grades for Flexibility." *Interface Age* 7, no. 4 (April 1982): 102, 104, 106, 108, 110. Description of CalcStar with brief mention of VisiCalc.

Smolin, Rocky. "Fin-Ratio: Financial Data and Ratios Program." *Interface Age* 7, no. 5 (May 1982): 100-102. Review of Fin-Ratio, marketed by FPS Assoc., including reference to VisiCalc.

"Softalk Presents the Best Sellers." *Softalk*, all issues. A department of this magazine in which monthly reports of the sales of software are presented.

Sojka, Deborah, and Philip H. Dorn. "Magic Moments in Software." *Datamation* (Special Report), August 25, 1981, 7-8, 10, 14, 16.

S P R E A D S H E E T. VisiGroup 1, no. 1 (November 1980, and subsequent issues). A valuable newsletter from VisiGroup (P.O. Box 1010, Scarsdale, NY 10583), National VisiCalc Users' Group. An index of articles from issues 1-7, 1981 appears in the January 1982 issue.

"Spread-Sheet Package For Use Under Unix OS: PDP-11 Installations To Get First Release." *Computer Business News* 4, no. 51 (December 21, 1981): 22. Description of Viewcomp from Unisoft Inc., with comparisons to VisiCalc.

Stinson, Craig. "Venture with VisiCalc." *Softalk* 1, no. 11 (July 1981): 14-17. (*Refer also* to Christopher U. Light, Frank Malone, Joe Shelton and Otto K. Wetzel entries in this bibliography.)

Strategies of Personal Computer Manufacturers. San Jose, CA: Strategic Incorporated, 1981. Report containing an analysis of VisiCalc.

Strothman, Jim. "Perspectives on Protection." *PC* 1, no. 2 (April-May 1982): 108-111.

"SuperCalc to the Rescue." *Popular Computing* 1, no. 2 (December 1981): 36. Describes features of SuperCalc not available in VisiCalc.

"Super Invader is Readers' Choice." *Softalk* 1, no. 8 (April 1981): 13. Results of Softalk's Most Popular Program Ever poll.

Tangorra, Joanne. "Software: Is the Shakeout At Hand?" *Popular Computing* 1, no. 6 (April 1982): 42-45. Discussion of the software industry including VisiCorp, Personal Software, which markets VisiCalc.

Tannenbaum, Michael. "80 Accountant." *80 Microcomputing* (May 1981): 44, 47. A comparison of Target and VisiCalc.

Thé, Lee. "The Power of Purchasing by Computer." *Personal Computing* 6, no. 2 (February 1982): 29-32, 148-150. Software use, including VisiCalc, by purchasing departments.

Tommervik, Allan. "The Great Arcade/Computer Controversy, Part 1: The Publishers and the Pirates." *Softline* 1, no. 3 (January 1982): 18-19, 21-22; ". . . Part 2: What Price Software?" 1, no. 4 (March 1982): 10-12. Sales of games are compared in terms of price and number sold with VisiCalc and other software.

UPTREND. Sunnyvale, CA: Personal Software Inc. 18 pages. A publication for users of the firm's software systems.

"Using VisiCalc: Building a Cash Flow Chart." *Farm Computer News* 1, no. 3 (July 1981): 6. A cash flow model for a farm.

Van Tassel, Dennie. *Program Style, Design, Efficiency, Debugging, and Testing*. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1974. As stated in the preface, this book is a survey of the lore of programming. Although written before VisiCalc was available, it's a good source book when read with a VisiCalc perspective.

"VisiCalc Forecasting Package Provides Reports, Analyses." *Computer Business News* 5, no. 20 (May 17, 1982): 18. Announcement of VisiCorp's Business Forecasting Model spreadsheets which use VisiCalc and the data interchange format, DIF to transfer data from one sheet to another.

"VisiCalc - Where the Action Is." *Computer Career News* 2, no. 6 (April 20, 1981): 10; and a related letter to the editor 2, no. 8 (May 18, 1981): 6.

VisiNews. 1, no. 1 (March 1981). Newsletter from Twenty-First Century Media Inc., P.O. Box 341, Kings Park, NY 11754.

Wetzel, Otto K. "Ventures with VisiCalc." *Softalk* 1, no. 10 (June 1981): 26, 28. Correlating and plotting utility bills using weather data. (*Refer also* to the Christopher U. Light, Frank Malone, Joe Shelton, and Craig Stinson entries in this bibliography.)

"What is this VisiCalc, Anyway?" *Personal Computing* 6, no. 2 (February 1982): 154. Brief description of VisiCalc capabilities.

Wiegner, Kathleen K. "Tomorrow Has Arrived." *Forbes* 129, no. 4 (February 15, 1982): 111-115, 119. Growth of the personal computer marketplace including mention of VisiCalc.

Williams, Robert E. and Brian L. King. *The Power of VisiCalc Volume II.* Portland, OR: Managment Information Source, 1982, 97 pages.

Williams, Robert E. and Bruce J. Taylor. *The Power of VisiCalc.* Portland, OR: Management Information Source, 1981, 88 pages.

Wise, Deborah. "Accountant Home-Finance Package Abjures Jargon." *InfoWorld* 4, no. 21 (May 31, 1982): 1. Describes the Accountant Finance Data Base System, from Decision Support Software, a double-entry accounting package which interfaces with VisiCalc.

Wise, Deborah. "$99.95 Timex Sinclair 1000 Microcomputer Debuts." *InfoWorld* 4, no. 18 (May 10, 1982): 3. Description of this product with a brief mention of a VisiCalc-type package to be available.

Wise, Deborah. "IBM Personal Computer Software Provides 4 Functions." *InfoWorld* 4, no. 20 (May 24, 1982): 13. Discussion of the MBA product, from Context Management Systems, in relation to VisiCalc.

Wise, Deborah. "Lawyers & CPA's Timidly Hop Onto Micro Bandwagon." *InfoWorld* 3, no. 32 (December 28, 1981): 14-15. Use of microcomputers, including VisiCalc use, by lawyers and CPA's.

Wise, Deborah. "Two Programs from ISM Offer Magical Aid to Nonprogrammers." *InfoWorld* 4, no. 17 (May 3, 1982): 8. Discussion of Graphmagic and MatheMagic and their interface/relationship to VisiCalc and VisiPlot.

Yasaki, Edward K. "A House is Still a Home." *Datamation* 28, no. 2 (February 1982): 74, 79-80, 82. Discussion of one-product software companies.

Zander, Ben. "Commodities Trading: The Computer Exchange." *Personal Computing* 6, no. 5 (May 1982): 82-83, 85. Microcomputer use in commodities trading, including VisiCalc.

Zussman, John Unger. "Battle of the Spreadsheets." *Infoworld* 4, no. 10 (March 15, 1982): 20-21; and a related letter to the editor, "On Target," from Les Trachtman 4, no. 14 (April 12, 1982): 20. Discussion of several spreadsheet programs.

Zussman, John Unger. "Hey, Look Who's Here: It's Personal Chapstick." *InfoWorld* 4, no. 21 (May 31, 1982): 32-34. Discusses company names in the computer industry, including the name change from Personal Software to VisiCorp.

Zussman, John Unger. "A Plea For Self-Documentation." *InfoWorld* 4, no. 16 (April 26, 1982): 24-25. Begins with the premise that software users don't read manuals, and discusses the need for self-documentation. Cites difficulties of a technical writer who documents several VisiCalc features.

Zussman, John Unger. "VisiCalc PLUS from HP." *InfoWorld* 3, no. 4 (March 2, 1981): 14-15. Software 'report card' on VisiCalc PLUS giving it grades of excellent and good.

INDEX

Special Characters are in Alphabetical Order by the Name of the Character

THE VISICALC® BOOK
Atari® Edition
DONALD H. BEIL

Personal computer users and hundreds of businesses across the country are discovering a wide variety of uses for the new VisiCalc software program. The power of the software is that *you* design the model, so you are able to examine every step of your business operations. Once you have constructed the model, all of your "what if" questions will be answered in a matter of minutes, eliminating hours of manual calculating. If you are presently using VisiCalc on your Atari home computer, and want to learn more about its expanded uses, or are thinking about purchasing VisiCalc—this book is for you. It will show you how to build a model, enter your data, and begin to explore all of your "what if" questions about estimates, or any aspect of your business you want to explore. There are a large number of practice problems included in this book to help the user become a skilled user.

The Table of Contents includes:

- The VisiCalc Software Program
- The Computer on Which We Use VisiCalc
- The Uses for VisiCalc
- The Data We Enter
- System Users
- The Capabilities of a VisiCalc System
- The Power of VisiCalc
- Getting Started
- Commands
- Labels, Numbers, Formulas
- Built-In Functions
- Creating Templates
- Documentation
- What Out Client, Secretary, or Boss Needs to Know
- The Limitations of the VisiCalc System
- Practice Problems

Cover by Debbie Balboni

RESTON PUBLISHING COMPANY, INC.
A Prentice-Hall Company
Reston, Virginia

0-8359-8393-5