

---

# VISICALC<sup>®</sup> ADVANCED VERSION

---

INSTANTLY CALCULATING 'ELECTRONIC WORKSHEET'



VISICORP<sup>™</sup>  
PERSONAL SOFTWARE<sup>™</sup>

Manual Copyright © 1982 VisiCorp  
All Rights Reserved.

Program Copyright © 1979, 1982 Software Arts, Inc.  
All Rights Reserved.

Printed in the U.S.A.

Apple® is a registered trademark of Apple Computer Inc.  
Centronics® is a registered trademark of Centronics Data Computer Corp.  
Corvus™ is a trademark of Corvus Systems, Inc.  
DIF™ is a trademark of Software Arts, Inc.  
Keystroke Memory™ is a trademark of Software Arts, Inc.  
Paper Tiger™ is a trademark of Integral Data Systems, Inc.  
ProFile™ is a trademark of Apple Computer Inc.  
Qume® is a registered trademark of Qume Corporation  
Silentype™ is a trademark of Apple Computer Inc.

VisiCalc® is a registered trademark of VisiCorp  
VisiSchedule™ is a trademark of VisiCorp

Part Number: 10950-2210

---

# VISICALC<sup>®</sup> ADVANCED VERSION

---

INSTANTLY CALCULATING ELECTRONIC WORKSHEET

## **User's Guide for the APPLE III**

Program by  
Software Arts, Inc.

Manual Written by  
Rob Merges  
and  
Van Wolverton



**VISICORP<sup>™</sup>**  
PERSONAL SOFTWARE<sup>™</sup>



**VISICALC® Advanced Version****CHAPTER 1  
INTRODUCTION**

About This Manual . . . . .	1-1
How to Use This Manual . . . . .	1-2
About the Program . . . . .	1-3
Playing What-If . . . . .	1-5

**CHAPTER 2  
THE VISICALC® TUTORIAL****LESSON ONE**

The Screen . . . . .	2-3
The Status Area . . . . .	2-4
Other Screen Features . . . . .	2-4
VisiCalc Commands . . . . .	2-4
The Cursor . . . . .	2-5
Moving the Cursor . . . . .	2-5
Correcting Typing Errors . . . . .	2-5
Scrolling the Window . . . . .	2-5
Going Directly to a Cell . . . . .	2-7
The Help Command . . . . .	2-8
Correcting With the ESCAPE Key . . . . .	2-8
Writing on the Worksheet . . . . .	2-9
Formulas and Recalculations . . . . .	2-10
More on Labels and Values . . . . .	2-12
Erasing an Entry . . . . .	2-13
Using the Cursor to Refer to Cells . . . . .	2-14
VisiCalc Formulas . . . . .	2-15
Saving Your Work . . . . .	2-16
File Names . . . . .	2-17
Saving a Worksheet . . . . .	2-17
Loading a Worksheet . . . . .	2-18
Protecting Your Work . . . . .	2-18
Making Backup Copies . . . . .	2-19
Summary . . . . .	2-19

**LESSON TWO**

Entering Labels as Titles .....	2-21
Changing the Width of One Column .....	2-23
Changing the Width of All Columns .....	2-24
Precision of Displayed Values .....	2-24
Replicating — The Key to Building a Worksheet .....	2-26
Replicating a Column .....	2-28
Replicating References to Other Cells .....	2-29
Adding a Series of Values .....	2-31
Drawing Lines on the Worksheet .....	2-32
More on the Sum Function .....	2-35
Controlling the Format of Entries .....	2-35
Setting a Local Format .....	2-36
Replicating a Format .....	2-36
Creating a Budget Calculator .....	2-37
Printing the Worksheet .....	2-38
Summary .....	2-39

**LESSON THREE**

Entering a Multi-Column Label .....	2-42
Formatting Left and Right .....	2-44
Formatting Integers and Dollars-and-Cents .....	2-46
Splitting the Screen .....	2-48
Replicating a Row .....	2-50
Summary .....	2-52

**LESSON FOUR**

Displaying File Names .....	2-54
Inserting Columns and Rows .....	2-54
Deleting Columns and Rows .....	2-55
Selecting the Largest of Several Entries .....	2-55
Selecting a Value That Meets a Condition .....	2-56
Describing a Comparison Condition .....	2-57
Inserting Multiple Columns .....	2-59
Looking Up Values in a Table .....	2-59
Copying Columns With the Replicate Command .....	2-61
Making a Worksheet More Flexible .....	2-64
Changing an Entry on the Worksheet .....	2-65
Editing a File Name .....	2-66
Summary .....	2-66

**LESSON FIVE**

Hiding an Entry . . . . . 2-68  
 Setting Tab Positions . . . . . 2-69  
 Controlling the Type of Data That Can Be Entered . . . . . 2-71  
 Protecting Cells Against Change . . . . . 2-72  
 Displaying the Formulas of the Worksheet . . . . . 2-72  
 Summary . . . . . 2-74

**LESSON SIX**

Saving a Portion of the Worksheet . . . . . 2-76  
 Defining Your Own Commands . . . . . 2-78  
 Saving a DIF File Using a Keystroke Memory Sequence . . . . . 2-80  
 Deleting a File . . . . . 2-81  
 Resolving Forward References . . . . . 2-81  
 Using a Worksheet as a Data-Entry Form . . . . . 2-82  
 Using DIF Files to Create a Summary Report . . . . . 2-84  
 Calculating an Average . . . . . 2-85  
 Summary . . . . . 2-87

**LESSON SEVEN**

Using Keystroke Sequences for Common Worksheet Features . . . . . 2-88  
     Editing a Sequence Definition . . . . . 2-89  
     Defining a Sequence Usable Anywhere on the Worksheet . . . . . 2-89  
     Building a Worksheet Fast With Keystroke Memory  
     Sequences . . . . . 2-91  
 Replicating a Rectangular Area . . . . . 2-91  
 Moving Rows and Columns . . . . . 2-92  
 An Income Variance Report . . . . . 2-93  
 Setting Manual Recalculation . . . . . 2-95  
 Summary . . . . . 2-97

**LESSON EIGHT**

Finding the Future Value of a Series of Payments . . . . . 2-100  
 Finding the Present Value of a Series of Payments . . . . . 2-102  
 Finding the Payment of a Loan . . . . . 2-103  
 Finding the Future Value of a Fixed Amount . . . . . 2-104  
 Finding the Interest Between Two Dates . . . . . 2-105  
 Finding the Present Value of Unequal Payments . . . . . 2-106  
 Finding the Internal Rate of Return . . . . . 2-107  
 Summary . . . . . 2-108

**LESSON NINE**

A Cost-Estimating Worksheet .....	2-111
Modulo Arithmetic .....	2-114
Integer Arithmetic .....	2-114
Drawing Bar Graphs .....	2-117
Summary .....	2-118

**CHAPTER 3  
THE VISICALC® COMMAND REFERENCE**

Introduction .....	3-2
Attribute Command - /A .....	3-5
Deleting Attributes .....	3-10
The Expression Format Option - /AE .....	3-10
The Hide Option - /AH .....	3-13
The Mode Option - /AM .....	3-13
The Tab Option - /AT .....	3-15
The Label Option - /AL .....	3-16
The Value Option - /AV .....	3-17
The Default Option - /AD .....	3-19
Blank Command - /B .....	3-20
Clear Command - /C .....	3-21
The Clear All Option - /CA .....	3-22
The Clear Sheet Option - /CS .....	3-22
Delete Command - /D .....	3-23
Edit Command - /E or Control-E .....	3-27
File Names .....	3-29
Prefixes and Pathnames .....	3-30
Subdirectories .....	3-31
Scrolling .....	3-31
Format Command - /F .....	3-33
Default Format - /FD .....	3-35
General Format - /FG .....	3-36
Integer Format - /FI .....	3-37
Left-Aligned Format - /FL .....	3-37
Right-Aligned Format - /FR .....	3-38
Dollars-and-Cents Format - /F\$ .....	3-38
Graph Format - /F* .....	3-39
Repeating Labels - /F- .....	3-39
Assigning Complex Formats Easily - /F = .....	3-40

**CONTENTS**

Format to Attribute Conversions .....	3-41
The Functions-@ .....	3-42
Arguments .....	3-43
Arithmetic Functions .....	3-44
Rounding for Less Precision: @ROUND (v, precision) .....	3-44
The Modulo Function-@ MOD (n, m) .....	3-45
The Dot Product Function-@ DOT PROD (range1, range2) ..	3-45
Arithmetic Functions That Use a List of Arguments .....	3-46
Arithmetic Functions Without Arguments .....	3-47
Functions That Select Alternative Values and Labels .....	3-47
The @LABEL and @VALUE Functions .....	3-49
Calendar Functions .....	3-50
Converting Calendar Dates to Absolute Dates-@VMDY and @MDY .....	3-51
The Day Function-@DAY .....	3-52
The Month Function-@MONTH .....	3-53
The Year Function-@YEAR .....	3-53
The Hours, Minutes, Seconds Function-@HMS .....	3-53
The Hour Function-@HOUR .....	3-54
The Minute Function-@MINUTE .....	3-54
The Second Function-@SECOND .....	3-54
Financial Functions .....	3-54
Interest or Discount Rate-@RATE .....	3-56
Number of Periods-@PERIOD .....	3-57
Payment-@PMT .....	3-57
Future Value-@FV .....	3-58
Present Value-@PV .....	3-58
Internal Rate of Return-@IRR .....	3-58
Net Present Value-@NPV .....	3-59
Logic Functions .....	3-59
Trigonometric Functions .....	3-61
Global Command-/G .....	3-62
Attributes-/GA .....	3-63
Column Width-/GC .....	3-63
Single Column Width-/GCC .....	3-65
Order of Recalculation Option-/GO .....	3-66
Recalculation Priority Option-/GR .....	3-69
Format Option-/GF .....	3-69
Go To Command- > .....	3-70
The Help Facility-? .....	3-70
Help Facility Options .....	3-71

Insert Command-/I .....	3-72
Row Option-/IR .....	3-74
Column Option-/IC .....	3-75
Keystroke Memory Command-/K .....	3-76
The Define Sequence Option-/K = or CONTROL-K = .....	3-78
Using Pauses (^P) in Sequences .....	3-82
The Clear Option-/KC .....	3-84
The Edit Option-/KE .....	3-84
The Review Sequences Option-/K RETURN .....	3-86
Labels .....	3-87
Move Command-/M .....	3-90
Specifying the Rows or Columns to be Moved .....	3-91
Indicating Where the Rows and Columns Are to be Moved .....	3-92
Moving Rows .....	3-92
Moving a Column .....	3-95
Print Command-/P .....	3-96
Printing on the Printer-/PP .....	3-97
Changing Printer-Setting Options-/PS .....	3-98
Pagination .....	3-100
Sending Setup Strings .....	3-100
Printing the Worksheet in Sections .....	3-101
Creating a Print File-/PF .....	3-102
Diskette Files .....	3-102
Printing Worksheet Formulas .....	3-103
Printing Formats, Attributes, and Formulas .....	3-103
Repeating Label Command-/- .....	3-104
Replicate Command-/R .....	3-106
Copying a Single Cell .....	3-109
Creating a Column of Entries from One Entry .....	3-111
Making One Copy of a Column .....	3-112
Making Several Copies of a Column .....	3-113
Making Several Copies of a Row .....	3-114
Replicating a Block Once .....	3-115
Replicating a Block Several Times .....	3-116
Replicating Formulas .....	3-117
Options of the Replicate Command .....	3-121
Replicating Cell References in Formulas: All Relative or All No Change-(R and (N .....	3-121
Replicating Attributes or Cell Contents-(A, (C .....	3-122

**CONTENTS**

The Screen .....	3-123
Status Area .....	3-124
Entry Line .....	3-124
Prompt Line .....	3-124
Edit Line .....	3-124
The Window and Worksheet .....	3-125
Cells (Locations) .....	3-125
The Cursor .....	3-125
Moving and Pointing the Cursor .....	3-125
Typeahead .....	3-126
Correcting Mistakes .....	3-126
Storage Command-/S .....	3-127
Load Option-/SL .....	3-128
Save Option-/SS .....	3-129
Delete Option-/SD .....	3-131
Quit Option-/SQ .....	3-131
DIF™ Options-S#S and S#L .....	3-132
Save DIF™ Option-/S#S .....	3-132
Load DIF™ Option-/S#L .....	3-134
Creating Subdirectories: The /SC Command .....	3-135
Titles Command-/T .....	3-138
Horizontal Option-/TH .....	3-139
Vertical Option-/TV .....	3-139
Both Option-/TB .....	3-139
None Option-/TN .....	3-139
Values .....	3-140
References to Other Cells .....	3-141
Precision of VisiCalc Values .....	3-143
Order of Calculation in Formulas .....	3-143
Calculating Values on the Edit Line .....	3-143
Recalculation Order .....	3-145
Forward and Circular References .....	3-145
Version Command-/V .....	3-148
Window Command-/W .....	3-149
Horizontal Option-/WH .....	3-149
Vertical Option-/WV .....	3-151
One Window Option-/W1 .....	3-153
Synchronized Scrolling Option-/WS .....	3-153
Unsynchronized Scrolling Option-/WU .....	3-153

**APPENDIX A****OPERATING THE APPLE III**

What You Need . . . . .	A-1
The Keyboard . . . . .	A-1
Handling Diskettes . . . . .	A-2
Loading the VisiCalc Program . . . . .	A-3

**APPENDIX B****CONFIGURATIONS FOR THE VISICALC PROGRAM**

Configurations for the VisiCalc Programs . . . . .	B-1
Copying Program Files to Your ProFile™ Hard Disk . . . . .	B-4

**APPENDIX C****EXCHANGING FILES WITH DIF™  
(Data Interchange Format)**

The Save DIF™ Option-/S#S . . . . .	C-2
Load DIF™ Option-/S#L . . . . .	C-2
Using the DIF™ Format Within the VisiCalc Program . . . . .	C-3
The DIF™ Format . . . . .	C-4
Header . . . . .	C-5
Data Records . . . . .	C-5
End-Of-Data Records . . . . .	C-5
Programs that Work With the DIF™ Format . . . . .	C-6
Dumping a DIF™ File . . . . .	C-6
Printing a Worksheet from a DIF™ File . . . . .	C-7
Creating a DIF™ File . . . . .	C-8

**APPENDIX D**

**FORMULAS USED TO CALCULATE THE FINANCIAL FUNCTIONS**

The Rate Function-@RATE .....	D-2
The Periods Function-@PERIODS .....	D-2
The Payment Function-@PMT .....	D-2
The Present Value Function-@PV .....	D-2
The Future Value Function-@FV .....	D-3
Net Present Value Function-@NPV .....	D-3
The Internal Rate of Return Function-@IRR .....	D-3
Calculating @FV and @PMT with Partial Periods .....	D-3

**GLOSSARY/INDEX**

Glossary/Index .....	Index-1
----------------------	---------

**FIGURE**

**TITLE**

1-1. The Electronic Worksheet .....	1-4
2-1. Empty Worksheet .....	2-3
2-2. Go To Command Prompt .....	2-7
2-3. An Automatic Recalculation .....	2-11
2-4. Calculating a Formula .....	2-13
2-5. Income as a Percentage of Sales .....	2-16
2-6. Branch Office Budget .....	2-20
2-7. Title and Column Headings of Budget .....	2-22
2-8. Making One Column Wider .....	2-23
2-9. Columns too Narrow for Values .....	2-25
2-10. Replicate Command Prompt .....	2-27
2-11. The Process of Replication .....	2-29
2-12. Replicating a Relative Reference .....	2-30
2-13. Single and Double Lines on the Worksheet .....	2-33
2-14. Expanded Budget .....	2-34
2-15. Invoice .....	2-41
2-16. Shipping Addresses of Invoice .....	2-44
2-17. Entry Line of Invoice .....	2-48
2-18. Screen Split Vertically .....	2-49
2-19. Invoice With Several Items .....	2-50
2-20. Totals Block of Invoice .....	2-51

2-21.	Invoice With Lookup Tables . . . . .	2-53
2-22.	Examples of Maximum Function . . . . .	2-56
2-23.	If Function Examples . . . . .	2-58
2-24.	Horizontal and Vertical Lookup Tables . . . . .	2-60
2-25.	Lookup Table for Retail Price . . . . .	2-61
2-26.	Lookup Table for Discount Percent . . . . .	2-63
2-27.	Printing the Formulas of the Worksheet . . . . .	2-73
2-28.	Invoice With Data Area To Be Saved . . . . .	2-76
2-29.	Data Area To Be Saved as a Separate File . . . . .	2-77
2-30.	Sales Report Worksheet for Data Files . . . . .	2-84
2-31.	Summary Sales Report With One Data Row . . . . .	2-86
2-32.	Income Variance Report . . . . .	2-93
2-33.	Inaccurate Variance Report . . . . .	2-96
2-34.	Financial Calculator . . . . .	2-100
2-35.	Future Value and Present Value Calculations . . . . .	2-103
2-36.	First Cost-Estimating Worksheet . . . . .	2-112
2-37.	First Cost-Estimating Worksheet . . . . .	2-113
2-38.	Second Cost-Estimating Worksheet . . . . .	2-116
2-39.	Graph of Monthly Sales . . . . .	2-118
3-1.	Comparing the Format and Attribute Commands . . . . .	3-6
3-2.	Attribute Settings . . . . .	3-7
3-3.	Local, Global and Startup Settings . . . . .	3-8
3-4.	Worksheet Displayed in Expression Format . . . . .	3-12
3-5.	Mode Option Settings . . . . .	3-14
3-6.	Gutters and Alignment in Cell Displays . . . . .	3-16
3-7.	Deleting Columns . . . . .	3-24
3-8.	A Reference Before Deleting a Column . . . . .	3-25
3-9.	A Reference After Deleting a Column . . . . .	3-26
3-10.	Lookup Tables . . . . .	3-49
3-11.	Recalculation by Column (Correct Result) . . . . .	3-67
3-12.	Recalculation by Row (Incorrect Result) . . . . .	3-68
3-13.	Inserting Columns . . . . .	3-73
3-14.	Worksheet Before Inserting Row . . . . .	3-74
3-15.	Worksheet After Inserting Row . . . . .	3-75
3-16.	Defining a Keystroke Sequence . . . . .	3-77
3-17.	A Pause in a Sequence . . . . .	3-84
3-18.	Moving Two Rows Downward . . . . .	3-93
3-19.	Moving Two Rows Upward . . . . .	3-94
3-20.	Moving Two Columns . . . . .	3-95
3-21.	Printer-Setting Options . . . . .	3-99
3-22.	Printer-Setting Pagination . . . . .	3-100

**CONTENTS**

3-23.	Repeating Labels . . . . .	3-105
3-24.	Replicating One Cell Across a Row . . . . .	3-110
3-25.	Replicating One Cell Down a Column . . . . .	3-111
3-26.	Replicating a Column Several Times . . . . .	3-113
3-27.	Replicating a Row Several Times . . . . .	3-114
3-28.	Replicating a Block Once . . . . .	3-115
3-29.	Replicating a Block Several Times . . . . .	3-116
3-30.	Formula Replicated No Change . . . . .	3-118
3-31.	Formula Replicated Relative . . . . .	3-119
3-32.	Startup Screen . . . . .	3-123
3-33.	Forward References by Column and Row . . . . .	3-146
3-34.	Horizontal Windows . . . . .	3-150
3-35.	Vertical Windows . . . . .	3-152
A-1.	Apple III Keyboard . . . . .	A-2
A-2.	Inserting a Diskette . . . . .	A-3
A-3.	The VisiCalc Screen . . . . .	A-4



## INTRODUCTION

### ABOUT THIS MANUAL

This manual teaches you the VisiCalc® Advanced Version program and gives you a fast and complete reference to any detail about the program.

First, it teaches you the program in Chapter 2, The VisiCalc Tutorial. The nine lessons in Chapter 2 introduce you step-by-step to the commonly-used features of the program. As you progress, the lessons show you how to use the functions of the program to create worksheets and develop applications such as a budget calculator, an invoice, a sales report, an income variance report, and a financial calculator. In these lessons you will learn not only how to use the functions of the program, but also how to solve a few common real-life financial problems.

Second, it gives you a complete reference in Chapter 3, The VisiCalc Command Reference. The Command Reference is organized alphabetically by command for easy retrieval of information. You can look up any detail or find an answer to a specific question easily. The Command Reference features a Command Structure Chart at the beginning, explanatory examples on the commands and functions of the program, numerous descriptive illustrations, and complete information on all aspects of the program.

## How To Use This Manual

Although you will use this manual primarily as a tutorial and reference, you can save yourself time and use it more effectively if you follow these suggestions:

- If you are using VisiCalc for the first time, read this Introduction and Appendix A before proceeding to Chapter 2, The VisiCalc Tutorial. Read the lessons in Chapter 2 in order, and do them at the computer. By the end of Lesson 2 you should know enough of the program to begin experimenting on your own. If you get stuck, refer to the Help text in the program by pressing ? at any time, go to the lesson material, or go to the Reference for a complete and detailed explanation.
- If you are familiar with the VisiCalc program (any version), you can use this manual in different ways to learn the new features:
  1. You can load your Advanced Version (see Appendix A) and start experimenting immediately. The Advanced Version operates like previous versions (moving the cursor, entering labels or values, and selecting and executing commands). Use the Help Facility in the program (type ? at any time) and explore the new options. Try out the new Attribute command, the Keystroke Memory command, the financial functions, and the Move, Insert and Delete commands on your own or with the help of the examples in the Reference.
  2. For a more systematic approach, skim this introductory chapter and Lesson 1 of the Tutorial and go directly to Lesson 2. (If you need instructions on how to load the program, see Appendix A.) The new features of the Advanced Version are introduced through applications in the remaining lessons. You can use the Reference to read in depth about any command as you need it.

There are two worksheets stored on your VisiCalc program diskettes. These two worksheets are the worksheets you are instructed to create in lessons Four and Five. By loading the first worksheet (named Invoice2.VC), you can begin Lesson Five without first doing Lesson Four. By loading the second worksheet (named Invoice3.VC), you can begin Lesson Six without first doing Lesson Five.

The Appendices are for assistance on special topics, for example: Apple III basic operating instructions (Appendix A), using various printers (Appendix B), exchanging VisiCalc data files (Appendix C), and financial function formulas (Appendix D).

**INTRODUCTION**

A separate Pocket Reference Booklet is included with this manual. You can use this Pocket Reference at all times to remind you of various commands and their options.

Finally, a handy Glossary/Index is included at the end of this manual for defining all terms and finding all topics.

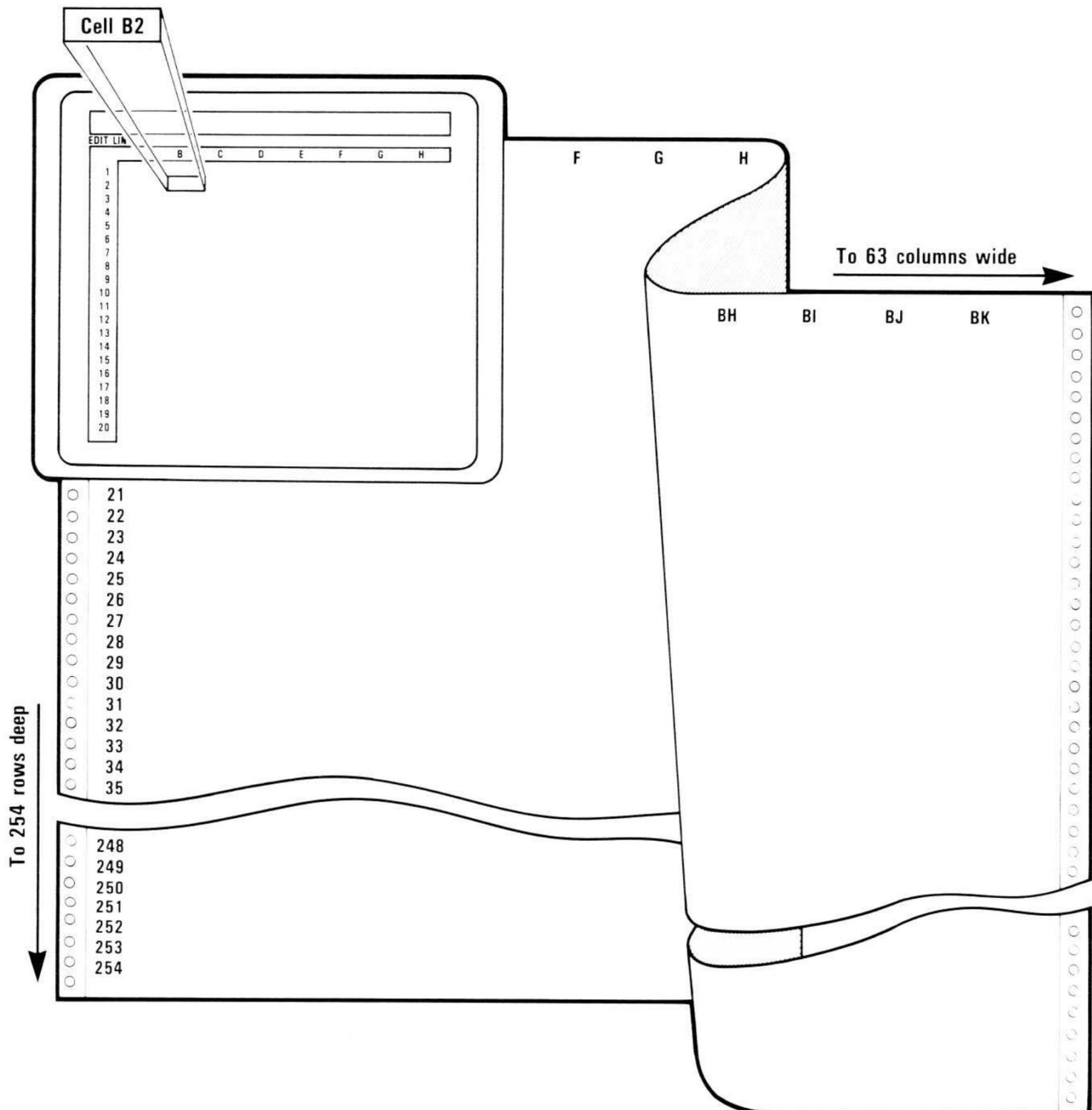
Use this manual as a flexible, complete guide to the VisiCalc Advanced Version program. The program itself gives you an ingenious and practical business tool for analyzing and solving corporate financial, planning, and budgeting problems.

**ABOUT THE PROGRAM**

The VisiCalc program is a proven problem-solver. Hundreds of thousands of professionals have used the VisiCalc program to plan budgets, project sales, figure income taxes, do profit-and-loss statements, solve engineering problems, estimate costs, balance accounts, figure income statements, manage costs, and account for all aspects of a large or small business.

The VisiCalc Advanced Version program lets you do more and lets you do it faster. You can now build your own often-used worksheets (templates) with a few "Keystroke Memory" sequences. You can store complex command sequences and use them any time for repetitive tasks to help you print or merge worksheets. You can format your worksheet to your taste with options such as variable column widths, variable decimal placements, entry protection, shielding entries, and centering entries. You can figure net present value, future value, internal rate of return, annuities and interest rates quickly. You can now move, copy, and reshape your worksheet with little effort. And you can use new arithmetic, calendar-figuring, or table lookup functions to meet your professional needs.

The VisiCalc Advanced Version is as easy to use and understand as previous versions. The program is built on the observation that many problems are commonly solved with calculator, pencil, and paper. However, adding rows or columns of figures with a calculator, writing the numbers on a paper worksheet, checking, erasing and recalculating the worksheet are pure drudgery. The VisiCalc program is a worksheet with rows and columns, but without the drudgery! It does the calculations automatically, using the power of the computer to do otherwise time-consuming tasks. It does much more; with the VisiCalc program, the computer screen becomes a window onto a large "electronic worksheet," as Figure 1-1 shows.



010-001

Figure 1-1. The Electronic Worksheet

## Playing What-If

The worksheet is organized as a grid of columns and rows. The intersections of the columns and rows define thousands of entry positions, each identified by its row-column coordinates. At each position you can enter a *value* — a formula that consists of numbers, arithmetic operators, functions, and references to other locations — or a *label* that describes some part of the worksheet.

By writing values on the worksheet you can set up charts, tables, and records; formatting commands let you control the appearance of each entry. You can make the VisiCalc worksheet look just like a checkbook register or a profit-and-loss statement.

But the power of the VisiCalc program lies in the fact that the computer *remembers* the formulas and calculations you use while working through a problem. If you change a value on the worksheet, all related values change as the VisiCalc program recalculates the entire worksheet.

Recalculation makes the VisiCalc program a powerful planning and forecasting tool. Not only can you easily correct mistakes, you can also explore alternatives. For example, suppose you're doing sales projections using the VisiCalc program. You may want to know what the impact on your company will be if a specific project doesn't sell as well as you had anticipated. What if you sell only 200 units a month instead of 250? What if you sell 300? What if one of your sales representatives quits and it takes six weeks to bring a new person up to speed? What is the value of a training program that brings new sales representatives up to speed in two weeks?

Playing what-if with the VisiCalc program is often a matter of changing a single value. If you enter a value at one location, you can copy it to any number of other positions. You can change, insert, delete, or move entire rows or columns. The worksheet is instantly restructured so that all relationships correspond to the changes. Doing the same thing with a calculator, pencil, and paper could take hours of erasing and recalculating.

The VisiCalc program can find the minimum, maximum, sum, average, and other calculations involving all or part of a row or column. Although the VisiCalc program automatically adjusts how values and titles are displayed to match your format commands, the full precision of values and length of titles remains in memory.

Once you've set up the worksheet to solve a particular problem, you can save it on disk and use the same setup for repeated instances or similar problems. All or part of the worksheet can be printed for reports or permanent records.

## THE VISICALC TUTORIAL

Chapter Two, the VisiCalc Tutorial, introduces the major features of the VisiCalc program in nine step-by-step lessons. The lessons are a keystroke-by-keystroke guide to be used at the computer. They require an initialized diskette.

**NOTE:** You cannot initialize a diskette while using the VisiCalc program, so you should initialize at least one blank diskette before you begin the lessons. See your *Apple III Owner's Guide* for instructions on how to initialize a diskette.

The lessons show you how to enter and use several real-life VisiCalc applications. To be as clear as possible, the manual uses the following conventions:

- When you are asked to type something, the exact characters you should type are printed in **boldface** type. "Type **Sales** →", for example, means type the word "Sales" followed by the right arrow key. However, type or press ESCAPE, RETURN, CONTROL, and TAB means press the ESCAPE key, the RETURN key, the CONTROL key, and the TAB key (do not type out these letters).
- When the manual refers to something displayed on the screen, the exact characters are printed in *Light* type. "The edit line reads +B1 - B2", for example, means the characters "+ B1 - B2" appear on the edit line.

**NOTE:** There are two worksheets stored on your VisiCalc program diskettes. These two worksheets are the worksheets you are instructed to create in lessons Four and Five. By loading the first worksheet (named Invoice2.VC), you can begin Lesson Five without first doing Lesson Four. By loading the second worksheet (named Invoice3.VC), you can begin Lesson Six without first doing Lesson Five.

## LESSON ONE

The Screen	2-3
The Status Area	2-4
Other Screen Features	2-4
VisiCalc Commands	2-4
The Cursor	2-5
Moving the Cursor	2-5
Correcting Typing Errors	2-5
Scrolling the Window	2-5
Going Directly to a Cell	2-7
The Help Command	2-8
Correcting With the ESCAPE Key	2-8
Writing on the Worksheet	2-9
Formulas and Recalculations	2-10
More on Labels and Values	2-12
Erasing an Entry	2-13
Using the Cursor to Refer to Cells	2-14
VisiCalc Formulas	2-15
Saving Your Work	2-16
File Names	2-17
Saving a Worksheet	2-17
Loading a Worksheet	2-18
Protecting Your Work	2-18
Making Backup Copies	2-19
Summary	2-19

---

In this lesson, you will get acquainted with the basic features of the VisiCalc program. You will learn how to move the cursor over the entire worksheet, make worksheet entries, save your work on diskette, and load data from the diskette. Load the VisiCalc program (if you're not familiar with the operation of the Apple III computer, see "Appendix A: Operating the Apple III Computer" for instructions). If the VisiCalc program is already loaded, type **/CAY** to clear the screen.

## THE SCREEN

The screen is now a *window* into the computer's memory. The VisiCalc program has divided it into rows and columns, producing an *electronic worksheet*. Figure 2-1 shows the empty VisiCalc worksheet. Rows are numbered 1, 2, 3, etc., and columns are lettered A, B, C, etc. Each intersection of a row and a column is a *location* or *cell*, referred to by its coordinates (A1, B3, C17, etc.). You can write a label, a number, or a formula at any location on the worksheet.

This electronic worksheet does many automatic calculations for you. It adds, subtracts, multiplies, divides, and performs more complex calculations at each location as you go. Both the number and complexity of calculations, and the layout and design of the worksheet are up to you.

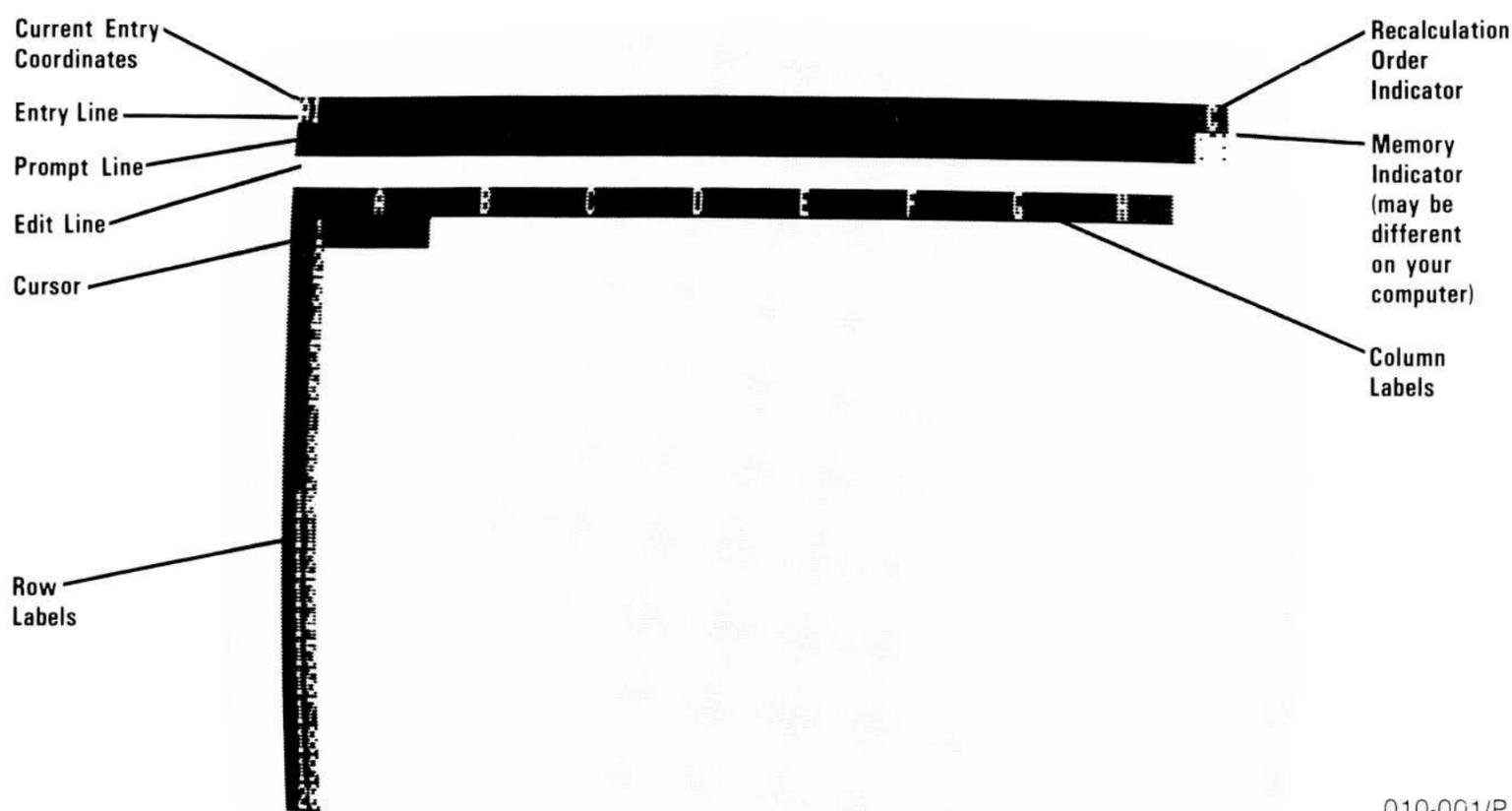


Figure 2-1. Empty Worksheet

## The Status Area

The light and dark bars at the top of the screen (just above the column letters) are the *status area*. The white bar is actually two lines: the top line is the *entry line* and the second line is the *prompt line*. The dark bar is the *edit line*.

The entry line shows what is entered at the location highlighted by the cursor. Right now the entry line shows A1, the location of the cursor.

The prompt line usually tells you what the VisiCalc program expects you to type next. At the moment, it displays the VisiCalc version number. If you call or write to VisiCorp with a question about the VisiCalc program, we need to know the version number and the model of your computer.

The edit line displays each character as you type it on the worksheet. You can change, or *edit*, this entry before you instruct the computer to write the label, number, or formula on the worksheet.

## Other Screen Features

The screen displays some other useful information. The letter in the upper-right corner of the screen tells you whether the VisiCalc program calculates new values across the rows (R) or down the columns (C). As you will see, this is an important choice. The number below it tells you how much memory is available for your worksheet; as you write on the worksheet, this number decreases. (Your number may be different from the number in the photo.)

Press the key marked RETURN and the version number disappears.

## VISICALC COMMANDS

Type /. The prompt line displays the letters that represent the VisiCalc commands:

Command: A B C D E F G I K M P R S T V W -

Each letter represents a VisiCalc command. As you follow the lessons, the commands will become familiar and natural. For the moment, don't worry what all the letters mean. The lessons teach you the name and function of each command. Remove the command list by typing **RETURN**.

## THE CURSOR

The intersection of column **A** and row 1 is location (cell) **A1**. A white rectangle, the *cursor*, covers it. Think of the cursor as the point where your electronic pencil meets the electronic worksheet. It marks where you can write. To keep track of where you are, the VisiCalc program displays the cursor's coordinate in the upper-left corner of the screen. Right now it reads **A1**.

### Moving the Cursor

You use the four arrow keys on your keyboard to move the cursor. The right-arrow key moves the cursor right one cell and the left-arrow key moves it left; the down-arrow key moves the cursor down and the up-arrow key moves the cursor up.

### Correcting Typing Errors

If, while you are practicing, you make a typing error, words may appear on the prompt line, the edit line, or the worksheet. Don't worry. Just type **CONTROL-C** (hold down the key marked **CONTROL**, and type **C**). Try it. Type **100**; now press **CONTROL-C**. The prompt and edit lines clear. The sound when you type **CONTROL-C** is the VisiCalc program's way of telling you that you canceled a command.

You can erase any entry written on the worksheet by positioning the cursor on it and typing **/B RETURN** (do not type out **RETURN**; press the **RETURN** key). You can clear the entire worksheet by typing **/CAY**. These commands are explained in detail later in this lesson, but might come in handy if you accidentally type a wrong key.

### Scrolling the Window

You can move the cursor to any location on the worksheet by scrolling. When you load the VisiCalc program, the cursor is at **A1**, the upper-left corner of the worksheet. Type **→** until the cursor is at the right edge of the screen. Type **→** again. The next column to the right comes into view, and column **A** disappears off the left edge. (You can verify this by watching the column letters just above the worksheet.)

The window has *scrolled* to the right. Type → a few more times, watching more columns appear at the right edge of the window while others disappear at the left.

The window also scrolls to the left. Type ← until the cursor is at the left edge of the window. Now type ← several more times; the columns that disappeared as you scrolled the window to the right come back into view. Type ← until the cursor is back at A1.

Type ← once more. The sound is the VisiCalc program's way of telling you that you're bumping into an edge of the worksheet. Now type ↑. This time you hear the sound because you're bumping into the top edge of the worksheet. So far you have encountered the left edge and the top edge of the worksheet. There are two other edges.

Type ↓ until the cursor moves to the bottom of the window. Type ↓ once more. The next row comes into view, while row 1 disappears off the top of the window. Type ↓ a few more times. As you can see, the worksheet is much larger than a sheet of paper.

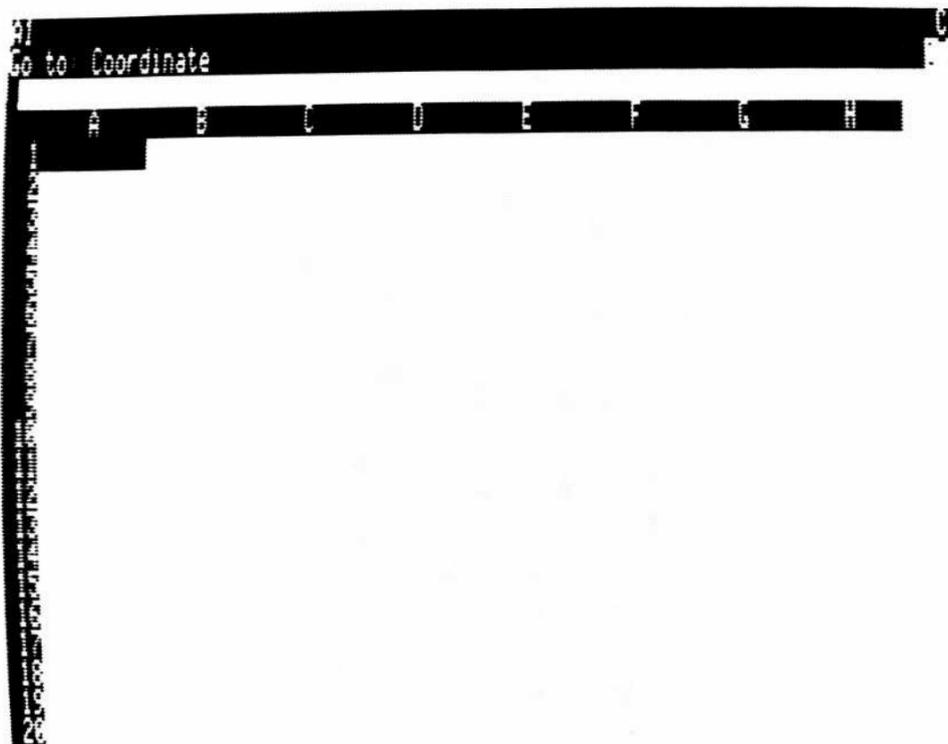
You can scroll more quickly by using your computer's automatic repeat feature. To see this in action, first press the ↓ key and *hold it down*.

The cursor and window scroll downwards automatically. To make the cursor scroll more quickly, press ↓ more firmly. Your computer's arrow keys have two automatic repeat speeds; pressing more firmly moves the cursor faster. Continue to hold down the arrow key until the cursor bumps into the bottom edge of the worksheet: it has reached location A254.

To move the cursor to the right edge of the worksheet, hold down →. The cursor scrolls to the right. Notice how succeeding columns are lettered: A through Z, then AA through AZ, and finally BA through BK. You are at the lower-right corner.

## Going Directly to a Cell

Even with automatic repeat, it takes a while to scroll to the lower-right corner. There's a much easier way to move the cursor to any cell: the Go To command. Type > (hold down the [SHIFT] key, and press the key marked with a period and the greater-than sign). Two things happen: Go To: Coordinate appears on the prompt line, and the edit cue (the small white square) appears directly below on the edit line. The screen should look like Figure 2-2.



010-002/P

Figure 2-2. Go To Command Prompt

Each time you type a command, the prompt line tells you what it expects you to type next. In Figure 2-2, the prompt line recognizes the Go To command (>) and asks you to type the coordinate. Type **B2**. The edit line reads B2, followed by the edit cue. Nothing has happened yet. The VisiCalc program doesn't know yet whether you want to go to position B2, or B21, or B215. Press **RETURN**. Now the prompt and edit lines clear, and the cursor moves to B2.

## THE HELP COMMAND

The VisiCalc program includes a description of its commands, functions, and operating procedures. You can display this information any time by typing a question mark (?). The program displays information related to what you are doing at the moment, and gives you the option of displaying more information or returning to the worksheet.

Try it. Type ?. The program reads the information from the diskette, and displays a description of what you can do now (at what it calls the "main level" of the program). After you read it, follow the prompts at the bottom of the screen. To get back to the worksheet, press **ESCAPE**.

To see how the Help command displays information related to what you are doing, type >?. The > is the Go To command; after you type it, the program is ready for you to type the coordinate of where you want the cursor to be moved. The Help command now displays information about moving the cursor. When you are ready, press **ESCAPE** to return to the worksheet. Press **ESCAPE** again to cancel the Go To command.

Whenever you aren't clear about what you should do next, or would like some information about a command, type ? to display the information.

## CORRECTING WITH THE ESCAPE KEY

Earlier, you pressed CONTROL-C to cancel something you typed. The VisiCalc program has a less drastic way of correcting errors — the **ESCAPE** key, which functions like the backspace key on a typewriter. It is represented in this manual with **ESCAPE**.

Type >**C11**. Suppose you intended to move the cursor to position C1, but accidentally typed 1 twice. The edit line now reads C11, followed by the edit cue. Press **ESCAPE**. The edit cue backs up one character and erases the extra 1, leaving C1. Now press **RETURN**. The cursor moves to C1.

In general, you can correct typing errors by backing up with **ESCAPE**; each time you press **ESCAPE**, the last character on the edit line is erased. Type >**A11**. Now press **ESCAPE** twice, leaving just A; then type 2 to get A2 and **RETURN** to end the command. The cursor moves to A2.

You can also cancel a command with **ESCAPE**. Type **>B5**. Suppose you change your mind and decide you don't want to move the cursor. Press **ESCAPE**, and the 5 disappears from the edit line. Press **ESCAPE** again; the B disappears from the edit line and **Go To: Coordinate** disappears from the prompt line. You have canceled the **Go To** command by erasing everything on the edit line; this accomplishes the same thing as typing **CONTROL-C**.

Before going on, spend a few minutes moving the cursor around with the arrow keys and the **Go To** command (**>**). Remember, in the remaining lessons, the entry instructions indicate with an arrow which way to move the cursor.

## WRITING ON THE WORKSHEET

As you have seen, moving the cursor and scrolling the window are fairly easy, but so far your worksheet is (or should be) empty. To restore the worksheet to the same condition it was in when you first loaded the VisiCalc program, type **/CAY** (the **Clear** command). The **Clear** command erases the worksheet, and positions the cursor at **A1**.

For your first entry, type **Sales** (but not **RETURN**). Look at the prompt line; it reads *Label*. A *label* is a worksheet entry that isn't used in calculating. The edit line reads *Sales*, followed by the edit cue. *Sales* also appears under the cursor at **A1**.

Now press **RETURN**. The prompt and edit lines clear, and **A1** reads *Sales*. The entry line reads:

```
A1 [L] Sales
```

The entry line describes the cell highlighted by the cursor: **A1** is the cell coordinate, **[L]** identifies the entry as a **Label**, and *Sales* is the entry.

As you type a label, it appears both on the edit line and on the worksheet under the cursor. It is not actually written on the worksheet, however, until you press **RETURN**. If you cancel the entry before pressing **RETURN**, what you have typed is not written on the worksheet.

Try it. Type **XXX**. The X's appear on the edit line and replace *Sales* at A1. Now cancel the entry by pressing **CONTROL-C**. The prompt and edit lines clear, and *Sales* reappears at A1 because you never actually wrote **XXX** on the worksheet.

Move the cursor to B1 (type →), and type **100**. The prompt line says *Value*, the Visicalc program's term for a number or formula. The edit line reads 100, but nothing appears at B1. When you enter a value, everything happens on the edit line until you end the entry. Now press **RETURN**. The prompt and edit lines clear, and 100 appears at B1.

Again, the entry line (the top line of the screen) describes the cell highlighted by the cursor:

B1 [V] 100

B1 is the coordinate, [V] means *Value*, and 100 is the entry.

Try pressing **ESCAPE**. Nothing happens. Once you have written a value on the worksheet, you cannot change it with **ESCAPE**. You'll learn how to change an entry a bit later.

## Formulas and Recalculation

Move the cursor to A2, and type **Cost**; then type →. The cursor moves to B2, leaving *Cost* at A2. You can write an entry on the worksheet not only by pressing **RETURN**, but also by typing an arrow key. The entry is written on the worksheet, and the cursor moves to the next location (as determined by which arrow key you type). The key you use to end an entry depends on which way you want to move the cursor (if at all) after writing the entry.

Assume that the cost of producing a product is 60 percent of sales. You need a formula at B2 that performs that calculation. Sales is 100 at B1, so the required calculation is .6 times 100. Because sales might change, however, the VisiCalc program lets you refer to the *location* of sales rather than the absolute value. The VisiCalc symbol for multiplication is the asterisk (\*); type **.6\*B1**.

The edit line reads  $.6*B1$ . (If it doesn't, use **ESCAPE** to correct any error.) The formula tells the VisiCalc program to multiply the value of the entry at B1 by .6. Now press **RETURN**. The prompt and edit lines clear, and the entry line reads B2 (V)  $.6*B1$ . B2 reads 60, the result of multiplying .6 times 100, the number at B1. The screen should look like Figure 2-3.



010-003/P

**Figure 2-3. An Automatic Recalculation**

To see the power of the VisiCalc program, move the cursor to B1. What if sales increase to 200? Type **200 RETURN**. The value of Cost is recalculated; B2 now reads 120, the result of multiplying .6 times 200. Whenever you change Sales at B1, Cost at B2 changes to .6 times the new value of Sales.

This automatic recalculation is one of the fundamental strengths of the VisiCalc program. You're not writing on paper, so you needn't worry about erasing entries that depend on other entries. You're not calculating by hand, so you needn't worry about making arithmetic errors or forgetting to change a figure that depends on another entry. The VisiCalc program does all this recalculation for you, automatically, whenever you change an entry on the worksheet.

## More on Labels and Values

You've calculated cost as a percentage of sales. What about income? That's sales minus cost, so you need a formula that subtracts cost (B2) from sales (B1).

First, label the row. Move the cursor to A3, and type I. The prompt line says *Label*. When you start typing, the VisiCalc program looks at the first character to determine whether you are typing a label, a value, or a command:

- If you start with A through Z, as you did here, it assumes you are typing a label.
- If you start with 0 through 9, a decimal point (.), a parenthesis (()), or something that could begin a formula — such as a plus or minus sign (+ or -) — it assumes you are typing a value.
- If you start with a slash (/), it assumes you are typing a command.

Cancel what you have typed with **CONTROL-C**.

What if you want to write a label such as 1983 Income or [Loss]? Try to type **1983 Income**. The VisiCalc program looks at the first character (1) and assumes you are typing a value. When you press the space bar after typing 1983, it beeps an error and refuses further input. You typed something that can't be part of a value (the space). Cancel this entry with **CONTROL-C**.

You can begin a label with a number, or any other symbol that the VisiCalc program interprets as the beginning of a value, by typing a quotation mark (") as the first character (don't type it yet). The VisiCalc program takes the " to mean that you want to type a label regardless of what the next character is. The quotation mark is not displayed, and does not become part of the label.

Now type ". As soon as you type ", the prompt line reads *Label*. Type **1983 Inc.** followed by → to end the entry. The cursor moves to B3, leaving 1983 Inc. at A3.

You wanted a formula that subtracts cost from sales, so with the cursor at B3, type **B1 - B2 RETURN**. But B3 reads B1 - B2, not 80 (the difference between B1 and B2). Look at the entry line: it reads B3 (L) B1 - B2. The (L) means the entry is a label. Labels don't calculate anything; if the location of a label is referred to in a formula, its value is 0. B1 - B2 is a perfectly legitimate formula. What happened?

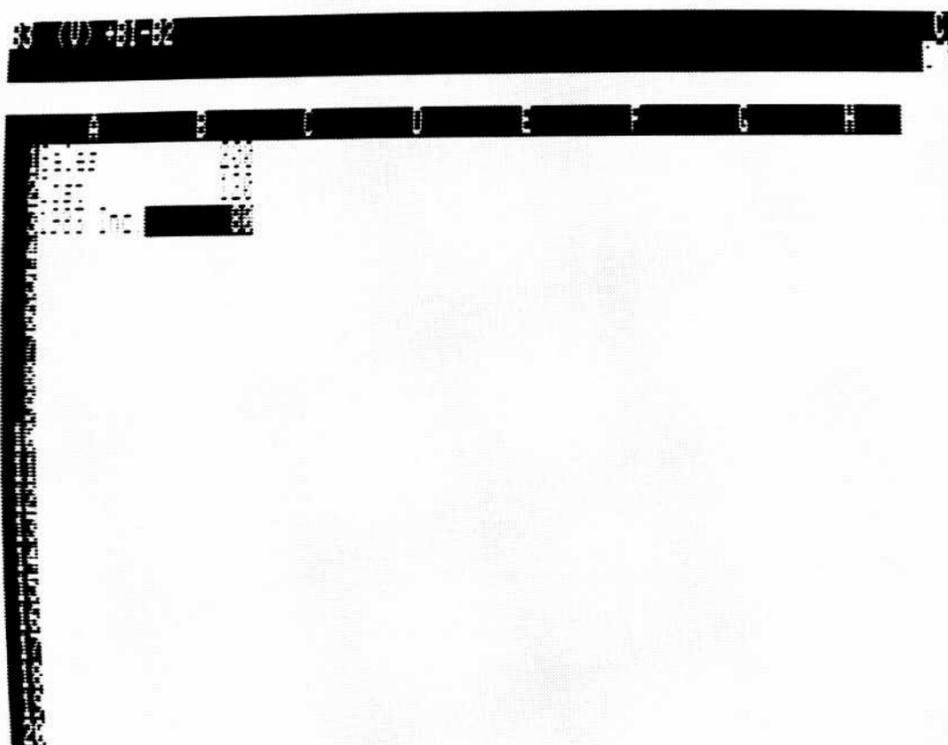
It begins with a letter. The VisiCalc program assumed you were entering a label.

### Erasing an Entry

Start over by erasing the label B1 – B2. Type /B (the Blank command). The prompt line reads Blank. If you press RETURN or an arrow key, the entry B3 is erased; if you type anything else, nothing happens. Press RETURN. The entry under the cursor disappears, and the entry line clears except for the coordinate B3.

Remember that you used " to type a label when the VisiCalc program would otherwise have treated it as a value? You can use a similar technique to make the VisiCalc program accept B1 – B2 as a value: start the formula with any character the VisiCalc program interprets as a value.

The formula + B1 – B2 is the same as B1 – B2; type +. The prompt line reads Value, because the VisiCalc program treats the first character (+) as a value. Complete the formula by typing B1 – B2 RETURN. Now B3 reads 80, the result of calculating + B1 – B2 (+ 200 – 120), and the entry line reads B3 (V) + B1 – B2. The screen should look like Figure 2-4.



010-004/P

**Figure 2-4. Calculating a Formula**

## Using the Cursor to Refer to Cells

As you wrote the formula for sales minus cost, you may have checked the screen to see if B1 and B2 were the correct locations. Imagine how much checking you could do on a worksheet filled with formulas.

Keeping track of coordinates can be time-consuming and confusing. If you scroll columns A and B off the screen, you can't see those locations. This makes it difficult to write a formula that refers to cells in those columns. The VisiCalc program gives you an easy way to solve this problem: as you write a formula, you can include the coordinates of another cell by moving the cursor to the location.

Type **+**. At this point you could type B1, as you did before. Instead, move the cursor to B1. The edit line reads **+B1**, the first part of the formula you want, just as if you had typed it.

Type **-** (the hyphen, for a minus sign). The cursor moves back to B3, the location of the formula you're entering. The edit line reads **+B1-**. As a rule, after moving the cursor to the cell to be included in a formula, continue the formula by typing an arithmetic operator (**+**, **-**, **\***, etc.).

Now move the cursor to B2. The edit line reads **+B1-B2** — exactly the same formula you typed before. To write the formula on the worksheet as it appears on the edit line, type **RETURN**. The entry line still reads B3 (V) **+B1-B2**. B3 reads 80, the result of subtracting B2 from B1. Nothing has changed except the way you wrote the formula.

What if sales are 300? Move the cursor to B1 and type **300 RETURN**. B2 changes to 180 (.6 times 300), and B3 changes to 120 (300 - 180).

This technique of moving the cursor to enter coordinates in a formula, sometimes called *pointing*, is often the most effective way of using the VisiCalc program. Like other operating methods, it quickly becomes second nature. In many cases, you'll find that you don't even think in terms of coordinates; you can concentrate on the content of the worksheet, dealing with its elements in visual terms.

## VISICALC FORMULAS

You have written labels, numbers, and simple formulas on the worksheet. A label, as you have seen, is a series of characters that usually identifies some other entry or area of the worksheet. If it is referred to in a formula, its value is zero. A value consists of a number or a *formula* and is used in calculations.

A formula can include numbers, arithmetic operators, references to other locations, and VisiCalc functions. (Functions perform additional arithmetic operations, special financial and scientific calculations, and are described later.)

The *arithmetic operators* specify the calculations to be performed:

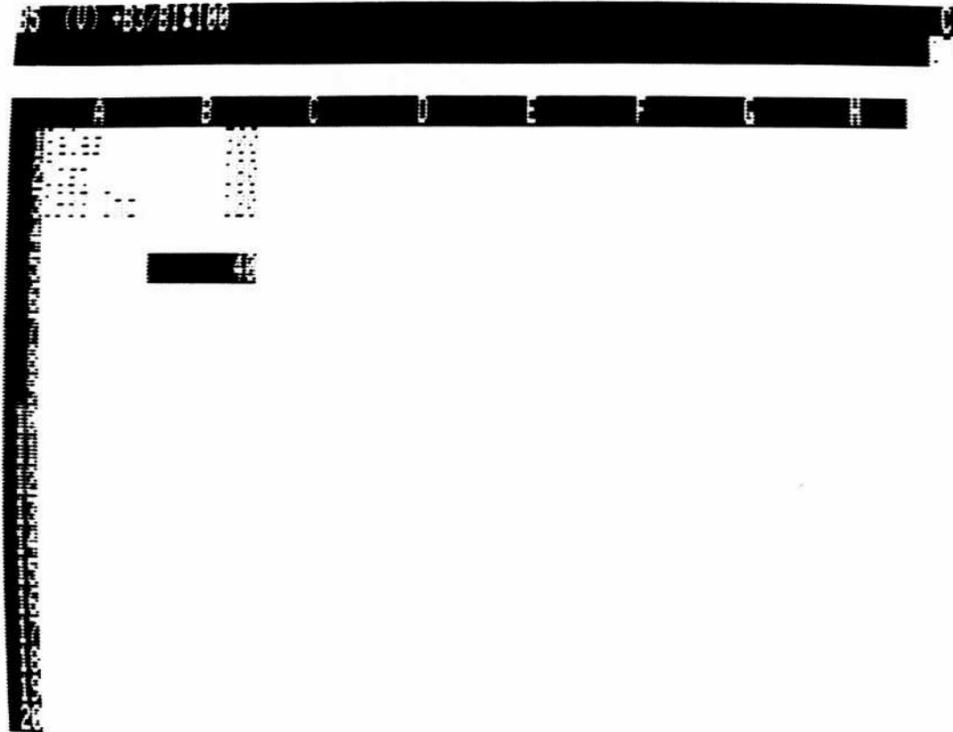
- + Addition
- Subtraction
- \* Multiplication
- / Division
- ^ Exponentiation

Arithmetic operations in a formula are performed from left to right; as in algebra, calculations in parentheses are performed first. The following are valid VisiCalc formulas:

- + F4
- + A20 - B3 + B4
- 1/K5
- 25 - C15
- + F3^2
- + B5\*B6\*H1
- + B3 - B4/D1
- + B3 - (B4/D1)

The last two produce quite different results. If B3 contains 30, B4 contains 15, and D1 contains 5, the next to last formula yields 3, but the last one yields 27 because the value at B4 is divided by the value at D1 before the result is subtracted from the value at B3.

Try it yourself. Move the cursor to position B5 and enter a formula that expresses income as a percentage of sales. Write this formula on the worksheet just by moving the cursor and typing +, /, \*, and **RETURN**. The screen should look like Figure 2-5.



010-005/P

**Figure 2-5. Income as a Percentage of Sales**

Hint: the formula is income divided by sales, multiplied by 100, or + B3/B1\*100.

## SAVING YOUR WORK

You can save the VisiCalc worksheet on an initialized diskette. *Diskettes cannot be initialized from within the VisiCalc program.* You will have to use one of the diskettes you prepared earlier. If you don't have any extra diskettes, you should still read the remainder of this lesson that describes how to save and load the worksheet.

## **File Names**

A saved worksheet is called a *file*. The name that identifies a file is called a *file name*. Because a diskette can hold many worksheets, you must give each worksheet on the same diskette a different file name. This way, you can find the worksheet later and load it into the computer.

There are a few simple rules for creating a valid file name:

- It can be up to 15 characters long.
- It can include only letters, numbers, and periods.
- The first character must be a letter.
- It cannot include a space or RETURN.

Your file name must have a prefix that specifies the number of the drive where you have placed your storage diskette. The prefix consists of a period (.), the letter D, the drive number, and a slash (/). To specify a file named BUDGET.VC in drive 2, for example, you would type .D2/BUDGET.VC.

When you specify a drive with a prefix, the VisiCalc program uses that drive for all subsequent disk operations unless you specify another. If you save a file named .D2/BUDGET.VC then later save a file named FORECAST.VC, the VisiCalc program tries to save FORECAST.VC on a diskette in drive 2. To save it on a diskette in drive 2, you must specify the drive with a prefix: .D2/FORECAST.VC.

**NOTE:** It doesn't harm anything to specify the diskette drive the VisiCalc program would use otherwise. If you save a file named .D2/BUDGET.VC and then save another a few minutes later with the name .D2/FORECAST.VC, everything works fine. Specifying the drive under these circumstances is merely redundant, not harmful.

## **Saving a Worksheet**

To save the worksheet, type /S. The prompt line reads Storage: L S D Q # C. Type S. The prompt line now reads Storage: File for saving, and the edit cue is on the edit line. Put the initialized diskette in drive 2.

The VisiCalc program is waiting for you to type the file name. Because this worksheet is an example, call it EXAMPLE. Type .D2/EXAMPLE to save the worksheet on the diskette in drive 2.

Now type RETURN. The disk drive begins whirring and the light comes on. After a moment the drive quiets down, the light goes off, and the prompt and edit lines clear. The worksheet is safely filed away on the storage diskette.

### **Loading a Worksheet**

To reassure yourself that the example worksheet was really saved, clear the worksheet (type /CAY). Type /SL (the Load option of the Storage command). The prompt line reads Storage: File to load.

Type **EXAMPLE RETURN**. Again, the disk drive begins whirring and, in a few moments, your worksheet is back on the screen. It remains on the disk until you erase it or replace it by storing another worksheet with the same name.

### **Protecting Your Work**

Have you ever worked out a problem or made some notes to yourself on a sheet of paper, only to find later that you lost the sheet or that someone accidentally threw it away? If you should lose power before you save the worksheet on a diskette, all the work you have done is lost.

To protect yourself, you should periodically save the worksheet on diskette. As you work, think how long it has been since you last saved the worksheet. If you have spent more time than you would care to lose if something goes wrong, or if you have results which might be difficult to reconstruct, then it's time to save the worksheet.

Printing the worksheet is another security measure. You can print not only the worksheet, but also the formulas and formats behind it. See "Print Command" and "Storage Command" in Chapter 3 for descriptions of how to print the content and structure of the worksheet. With printed copies of the worksheet and its underlying formats and formulas, you can reconstruct the worksheet without a disk file, if necessary.

## Making Backup Copies

Although a diskette is a safe and reliable medium for storing information, saving your work periodically is only the first step in protecting yourself. A diskette can be scratched or pick up grease and dust. It can be damaged by heat, exposed to a magnetic field, or accidentally initialized (which erases it). Eventually it wears out. The average lifetime of a diskette is about 40 hours of use (whenever the disk drive light is on, the diskette is in use). For complete protection, you should make backup copies of your files on separate diskettes. You can make a backup copy of an entire disk with the disk copy program on your Apple III Utilities Disk.

Use the Storage command to make a backup copy of a file. Insert the diskette that contains the file you want to copy into drive 2 (or whatever drive you are currently using). Then use the Load option of the Storage command (/SL) to load the worksheet. Remove the diskette, insert an initialized diskette, and use the Save option of the Storage command (/SS) to save the same worksheet on the second (backup) diskette.

## SUMMARY

This completes Lesson One. You may wish to experiment for a while, moving the cursor around and writing your own labels, numbers, and formulas. Try writing some formulas by pointing the cursor to specify the coordinates. Use the Help command to refresh your memory if you forget a detail.

# LESSON TWO

Entering Labels as Titles	2-21
Changing the Width of One Column	2-23
Changing the Width of All Columns	2-24
Precision of Displayed Values	2-24
Replicating — The Key to Building a Worksheet	2-26
Replicating a Column	2-28
Replicating References to Other Cells	2-29
Adding a Series of Values	2-31
Drawing Lines on the Worksheet	2-32
More on the Sum Function	2-35
Controlling the Format of Entries	2-35
Setting a Local Format	2-36
Replicating a Format	2-36
Creating a Budget Calculator	2-37
Printing the Worksheet	2-38
Summary	2-39

In Lesson One, you learned the basic operating techniques of the VisiCalc program. In this lesson you will create a typical budget — in this instance, for a branch office. Figure 2-6 is a printed copy of the budget.

By the time you finish this lesson, you will know enough about the VisiCalc program to create some worksheets of your own.

Budget  
1983 Est.

Salaries	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Total
Secretary	1000	1000	1000	1000	1000	1100	1100	1100	1100	1100	1100	1100	12700
Receptionist	800	800	800	800	800	800	800	800	800	800	800	800	9600
Accts. Clerk	850	850	850	850	850	850	850	850	850	0	0	0	7650
Office Mgr.	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250	15000
	3900	3900	3900	3900	3900	4000	4000	4000	4000	3150	3150	3150	44950
Phys. Plant													
Rent	1300	1300	1300	1300	1300	1300	1300	1500	1500	1500	1500	1500	16600
Utilities	200	200	200	200	200	200	160	160	160	160	160	160	2160
Insurance	400	400	400	400	400	400	400	400	400	400	400	400	4800
Maintenance	150	150	150	150	150	150	150	150	150	150	150	150	1800
Furniture	500	500	500	500	500	500	500	500	500	500	500	500	6000
	2550	2550	2550	2550	2550	2550	2510	2710	2710	2710	2710	2710	31360
	6450	6450	6450	6450	6450	6550	6510	6710	6710	5860	5860	5860	76310

**Figure 2-6. Branch Office Budget**

010-002

## **ENTERING LABELS AS TITLES**

Start by clearing the screen (type **/CAY**). To put a title for the budget at the top of the worksheet, type the following (remember, the arrow indicates the direction in which to move the cursor):

```
Budget ↓  
"1983 Est. RETURN
```

You need the quotation mark (") because the second line of the title begins with a 1; if you didn't type the ", the VisiCalc program would assume you were entering a value.

Now enter the column headings:

```
> A5 RETURN  
Salaries → Jan → Feb → Mar →  
Apr → May → Jun → Jul → Aug → Sep →  
Oct → Nov → Dec → Total RETURN  
> A7 RETURN
```

If you make a typing mistake, use **ESCAPE** to erase the last character. If you write an incorrect entry on the worksheet, just retype the entry (or erase it with the **Blank** command).

You have written the title and column headings on the worksheet; the screen should look like **Figure 2-7**.

Budget	1983 Est.	Jan	Feb	Mar	Apr	May	Jun	Jul
Salaries								

010-006/P

**Figure 2-7. Title and Column Headings of Budget**

You're ready to start entering budget line items. Each consists of a label in the first column (column A), and monthly figures in columns B through M. If the cursor isn't at A7, move it there, and enter the labels by typing:

**Secretary** ↓  
**Receptionist** ↓  
**Accts. Clerk** ↓  
**Office Mgr.** RETURN

A7 now reads *Secretary*, but A8 reads *Reception*. The remainder of the label isn't lost; its full text is stored in the computer's memory. The VisiCalc program initially sets up all columns with a width of 9, and displays only the first 9 characters of any label wider than that. Move the cursor to A8 (type ↑ ↑) and check the entry line (at the upper left of the screen). It reads:

A8 [L] Receptionist

A8 is the location, [L] identifies the entry as a label, and *Receptionist* shows that the full text you entered is, in fact, stored in the computer's memory.

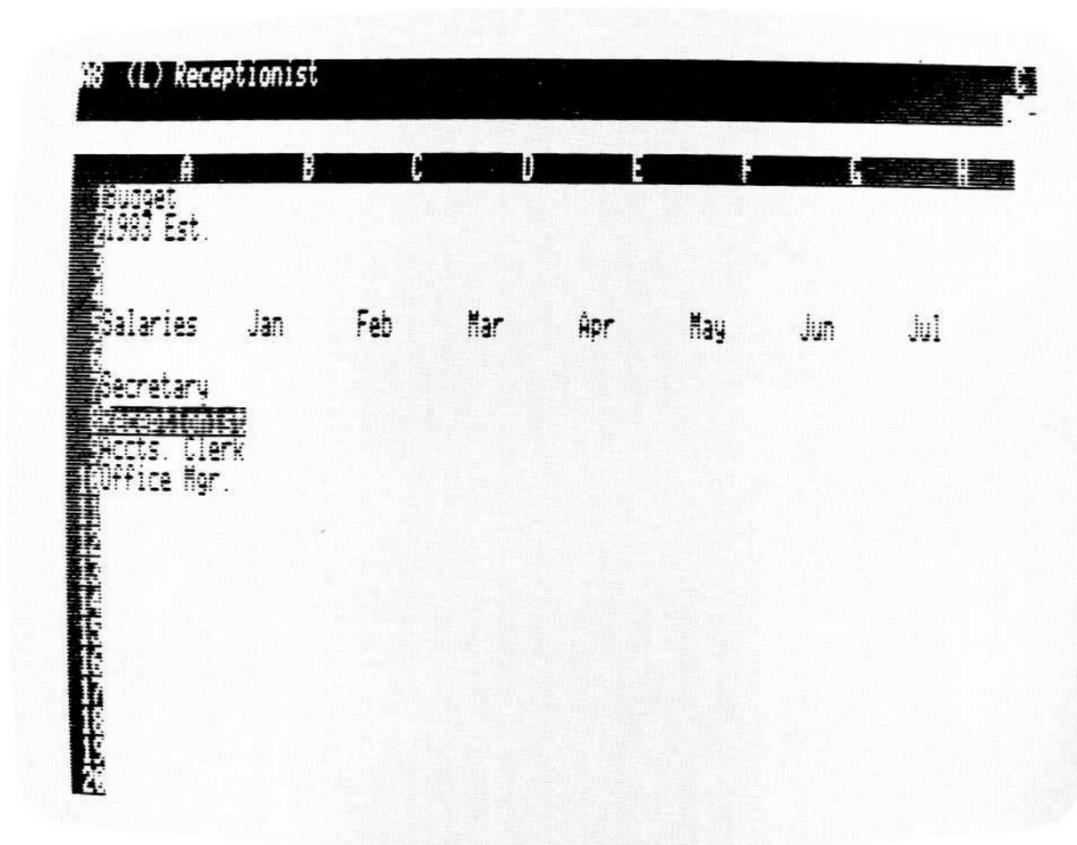
## CHANGING THE WIDTH OF ONE COLUMN

You can change the width of a column to display a wide label or value. Such a change is called a *global* change, because it affects more than a single location. Type **/G** (the Global command).

The prompt line reads: Global: A C O R F. The characters following Global: are options of the Global command. You want C, the column-width option. The other options are described later.

Type **C** to set the global column width. The prompt line reads Global: Column width: # or C [Current width: 9]. If you typed a number now, the width of all columns would be changed. You want to change the width of just one column, so type **C**. Now the prompt line reads Single column width: # or D [Current width: D].

You can set individual columns to any width from 1 to 125. The longest label you entered has 12 characters, so type **12 RETURN**. Now the entire label is displayed in Column A. The screen should look like Figure 2-8.



010-007/P

**Figure 2-8. Making One Column Wider**

Now to enter some values. Type the following to enter the January numbers:

**> B7 RETURN**  
**1000 ↓ 925 ↓ 850 ↓ 1250 RETURN**

## CHANGING THE WIDTH OF ALL COLUMNS

Besides changing the width of one column, you can change the width of all columns. It's often helpful to make the columns narrow so that more fit on the screen. To see this, type **/GC5 RETURN**.

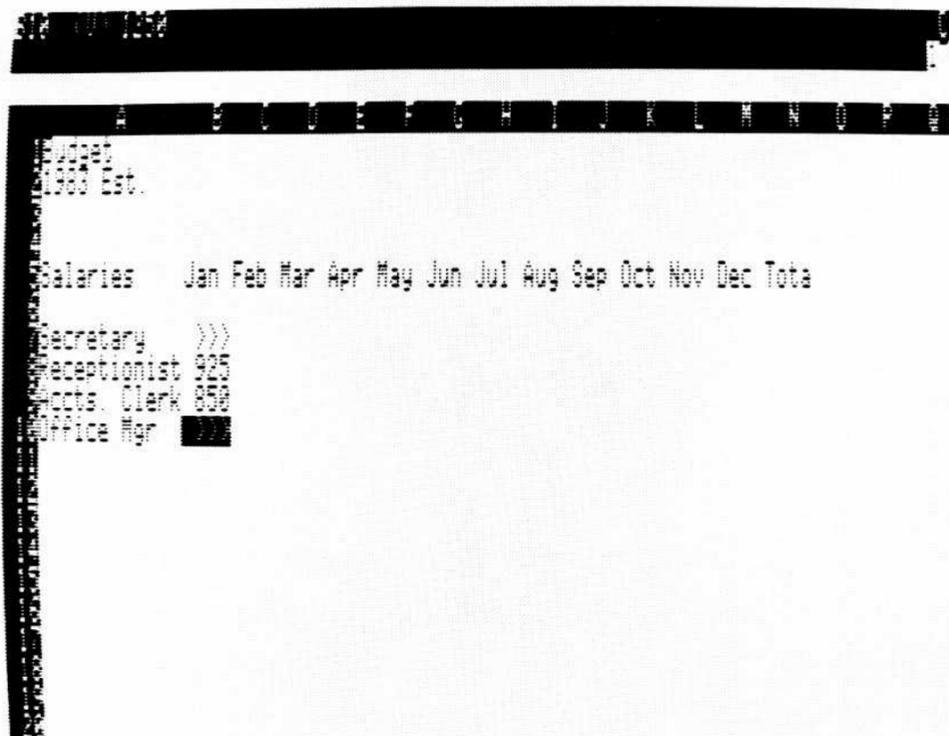
## PRECISION OF DISPLAYED VALUES

The VisiCalc program displays a number to the greatest precision possible in the space available. It leaves the first space of a column blank to keep columns of numbers from running into each other, so the space available to display a value is 1 less than the column width. What if there isn't room to display all the digits of a number?

Make the columns even narrower by typing **/GC4 RETURN**, and look at B7. Instead of 1000, it reads 1E3. This means "1 times 10 to the 3rd power," or 1000. Even though there isn't room for all four digits of 1000, it displays the value. But look at B10; instead of 1250 it, too, reads 1E3. That is the maximum precision possible in 4-character columns, so 1000 and 1250 are both displayed as 1E3.

What if there simply isn't room to display a number in the available space? If you specified that all these numbers were to be displayed as integers, it would require a column width of at least 5 to display them. Change the global format to Integer by typing **/GFI**. Now look at B7 and B10. They read >>>. If the VisiCalc program cannot display a value because the columns are too narrow, it fills its location with greater-than signs (>) to warn you.

The screen should look like Figure 2-9.



010-008/P

**Figure 2-9. Columns too Narrow for Values**

Type **/GFG/GC9 RETURN** to return to the normal column width and set the global format to General.

One last item completes the January entries: the monthly total. One way to do this is to add the numbers for each line item. Move the cursor to B12, and type the following:

- + ↑↑↑↑↑ (cursor moves to B7)
  - + ↑↑↑↑ (cursor moves to B8)
  - + ↑↑↑ (cursor moves to B9)
  - + ↑↑ (cursor moves to B10)
- RETURN**

The entry line reads **B12 (V) + B7 + B8 + B9 + B10**, and **B12** reads 4025. Now B12 always contains the total of the January expenses, no matter how you change the numbers. What if the Receptionist's salary were 800? Move the cursor to B8, and type **800 RETURN**. The total at B12 changes to 3900.

Again, here is an example of the power of automatic recalculation. No matter how many times you change the line items in the January budget, the total remains accurate.

But there are 12 months in the budget, and you have only entered the first month. You could enter each number for each month, but the VisiCalc program was designed to eliminate this sort of tedium. It gives you a much quicker way to write repetitive entries: the Replicate command.

## REPLICATING – THE KEY TO BUILDING A WORKSHEET

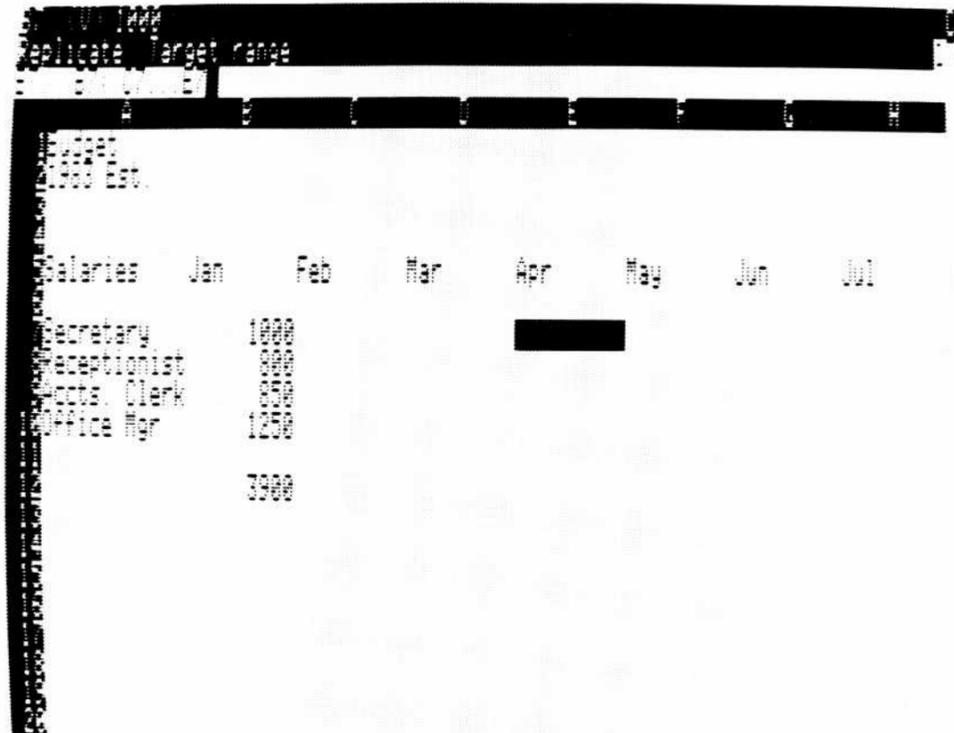
Automatic recalculation of the worksheet has been described as one of the fundamental strengths of the VisiCalc program. The Replicate command is another of those fundamental strengths. It replicates, or repeats, an entry or range of entries into as many other locations as you specify. The entry to be replicated can be any value or label.

To see how the command works, you can replicate the secretary's salary figure across all the months of the budget. Move the cursor to B7 and type /R. The prompt line reads Replicate: [, Source range or RETURN. This means the VisiCalc program wants you to specify what is to be replicated (the *source*). The edit line reads B7; this means that the source of the replication is B7, which is what you want to replicate. Press RETURN.

The edit line reads B7 . . . B7:, meaning that you have defined the source as cell B7 only. The ellipsis (. . .) separates the beginning of a range from the end of a range. The prompt line reads Replicate: Target range. This means the VisiCalc program now wants you to specify where you want the source replicated (the *target*).

Move the cursor to C7. The edit line reads B7 . . . B7: C7. If you just wanted to replicate B7 to C7, you would type RETURN. But you want to replicate B7 to all the columns from C7 (February) through M7 (December). C7 is the start of the target range, so type a period (.) to specify C7 as the beginning of the range. The cursor jumps back to B7, and the edit line reads B7 . . . B7: C7 . . .

Now to define the end of the target range. Move the cursor to D7. The edit line reads B7 ... B7; C7 ... D7. If you typed RETURN now, the 1000 at B7 would be replicated to C7 and D7. You're after a larger range than that, so move the cursor to E7 and check the edit line. The target range on the edit line is C7 ... E7. The screen should look like Figure 2-10.



010-009/P

Figure 2-10. Replicate Command Prompt

As you move the cursor across the row, watch the end of the target range change. Move the cursor to M7 and press RETURN.

The cursor jumps back to B7, and 1000 appears in each visible column of row 7. Move the cursor across the row, from B7 to C7 to D7, etc., continuing all the way to M7 (the December column). Each location in the row reads 1000, just as if you had typed it.

## Replicating a Column

You could use a Replicate command for each of the remaining line items to complete the salary expense portion of the budget. While that is much quicker than entering each item for each month, the Replicate command has more to offer. Remember, it asks for a source *range*. You can replicate all three remaining items with one command by including all of them in the source range.

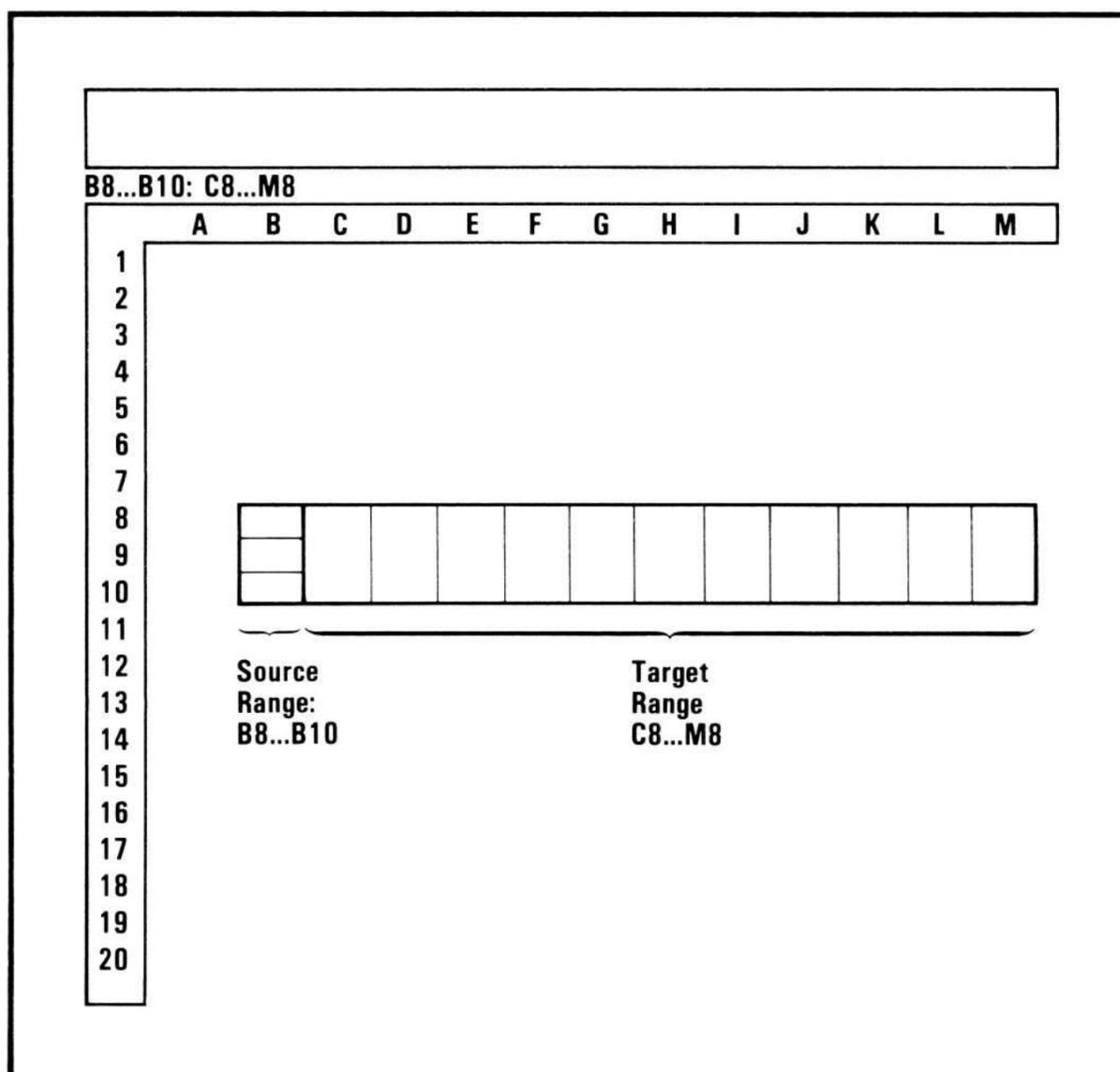
Move the cursor to B8 and type **/R**. Now move the cursor to B10. The prompt line reads Replicate: [, Source range or Return, and the edit line reads B8 . . . B10. Do you want to replicate the entries at B8, B9, and B10 across their rows? Yes. Press **RETURN**.

The cursor moves back to B8. Move the cursor to C8 (under 1000). The prompt line reads Replicate: Target range and the edit line reads B8 . . . B10: C8. If you pressed **RETURN** now, B8 would be replicated to C8, B9 would be replicated to C9, and B10 would be replicated to C10. C8 is only the beginning of the target range to which the source range is to be replicated, so type a period (.)

Move the cursor to D8. The edit line reads B8 . . . B10: C8 . . . D8. If you typed **RETURN** now, B8 would be replicated to C8 and D8, B9 would be replicated to C9 and D9, and B10 would be replicated to C10 and D10. Move the cursor to M8 and press **RETURN**. The cursor jumps back to B8, and numbers appear in all the visible columns. The values for all three line items have been replicated across to column M (December).

You can either use the cursor or type a coordinate to specify the source and target ranges in the Replicate command. In the previous Replicate command, it would have been quicker to type M8 after typing the period rather than moving the cursor to M8. You'll find yourself doing it both ways, depending on the size of your worksheet and whether you know the coordinates of the source and target ranges.

Figure 2-11 illustrates the process of replication, using the source and target ranges of the last example.



010-003

Figure 2-11. The Process of Replication

### Replicating References to Other Cells

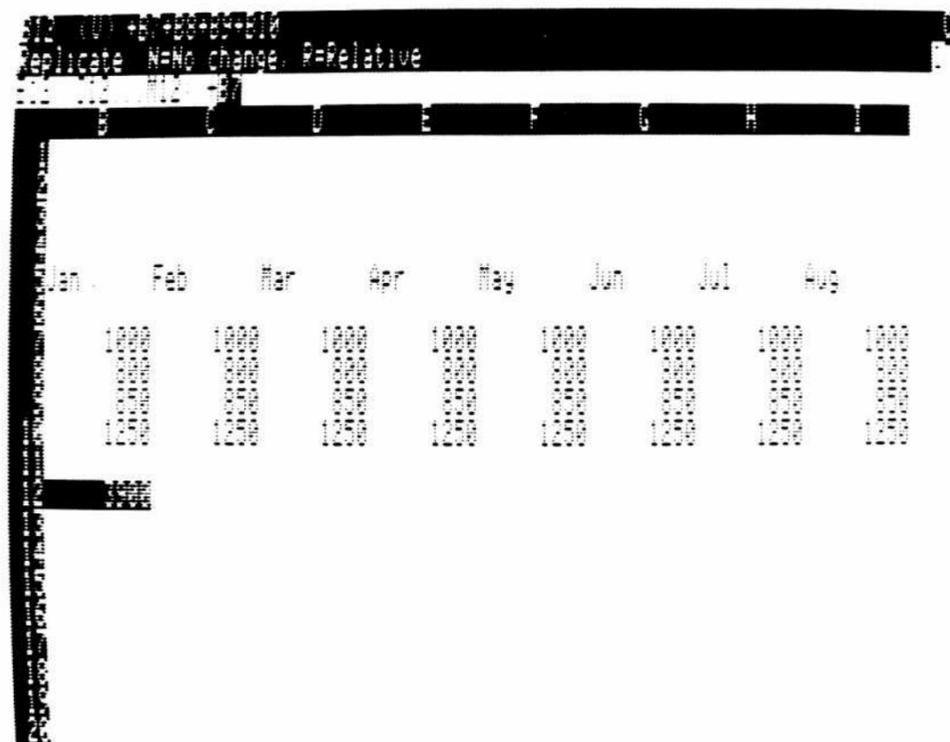
The budget needs a total for each month; again, you can use the Replicate command. Move the cursor to B12. The entry line reads B12 (V) +B7 +B8 +B9 +B10, and B12 reads 3900. Think about this formula. It adds each of the line items in column B (January). C12, the total for the February column, should add each of the line items for column C. D12, the total for the March column, should add each of the line items for column D, and so on through column M (December).

The formulas to add these columns are identical except for the column letter: +C7+C8+C9+C10 in column C, +D7+D8+D9+D10 in column D, etc. The Replicate command can make this change for you automatically as it repeats the formula.

Type the following:

**/R RETURN → . M12 RETURN**

The program isn't through yet this time, because it needs to know how to treat the references to other locations in the formula. (In the previous Replicate commands you were replicating only numbers.) The screen should look like Figure 2-12.



010-010/P

**Figure 2-12. Replicating a Relative Reference**

The edit line reads B12: C12 ... M12: +B7 and the edit cue highlights +B7. This gives you a complete picture of the Replicate command: it will replicate B12 (the source range) to C12 through M12 (the target range). The highlighting of +B7 is the VisiCalc program's way of asking you how it should treat the reference to B7 as it replicates the formula. You have two choices: N for No change, R for Relative. If you specify No change, each target location will refer to B7. If you specify Relative, the column letter will change in each target column: C7 in column C, D7 in column D, etc.

You want the coordinate to change in each target location, so type **R**. Now +B8 is added to the edit line, highlighted by the edit cue; you want this reference to be relative to the new location, too, so type **R** again. Each reference, in turn, is highlighted; each is to be relative, so type **R** for each one.

After you type the fourth R, the totals appear across the screen. The VisiCalc program is adding each column of numbers and displaying the total in row 12.

This last operation better illustrates the power of the Replicate command. You didn't simply copy the same number into several other locations, you copied a formula that contained references to other locations. Those references were changed in each new location to perform the same calculations relative to the new location. No matter which of the 48 budget numbers you change, the monthly totals remain accurate. This is a much more powerful planning tool than the same budget written on a piece of paper.

## ADDING A SERIES OF VALUES

To see how quickly you can add useful information to a worksheet, notice what is required to add yearly totals. Type **>N7 RETURN** to move the cursor to the yearly total position for the first line item. The monthly figures are in columns B through M, so the yearly total is the sum of B7 through M7. You could type **+B7+C7+D7+...** and so on but, again, the VisiCalc program comes to your aid.

In addition to commands, VisiCalc includes *functions*, a set of commonly needed calculations that can be used in formulas. Each function has a name, and usually requires one or more *arguments*, values upon which the function operates.

To add a series of locations, for example, there is the Sum function (typed **@Sum**; the name of each function starts with **@**). Its arguments specify what locations or values are to be added. **@Sum(3,5)**, for example, yields 8. **@Sum(B7,B8)** yields the sum of the values at B7 and B8. If the values of B7 and B8 change, **@Sum(B7,B8)** always yields their sum.

To make N7 (the yearly total) always contain the sum of B7 through M7 (the monthly totals), type **@SUM(B7.** and look at the edit line. As soon as you type the period, the VisiCalc program displays the ellipsis, indicating that B7 is the beginning of a range (just like in the Replicate command). Complete the function by typing **M7) RETURN**. Almost immediately the yearly total for row 7 appears at N7: 12000.

One Replicate command gives you all remaining yearly totals; type the following:

**/R RETURN ↓ . ↓ ↓ ↓ ↓ ↓ RETURN**

Now the formula appears on the edit line with B7 highlighted, asking whether the reference should be replicated with No change or Relative to the new locations. You want N8 to be the sum of B8 through M8, N9 to be the sum of B9 through M9, etc., so the references should be relative. (Don't worry about row 11 being blank; a formula that yields 0 will be placed at N11, but you'll get rid of it in a moment.)

Type **R**, and when M7 is highlighted, type **R** again. In a most satisfying way, totals appear in all the rows, and you have completed the salaries portion of the budget. You have a dynamic document; no matter how you change the line items, the vertical (monthly) and horizontal (yearly) totals remain accurate.

Try it. What if the secretary's salary (the first one) increases to \$1200 in December? Move the cursor to M7 and type **1200 RETURN**. The yearly total for that row changes to 12200, the monthly total for December changes to 4100, and the yearly total for all salaries changes to 47000.

## DRAWING LINES ON THE WORKSHEET

Lines are often used to separate column headings from the data. You could enter dashes all across the worksheet from A6 to N6, but the VisiCalc program gives you an easier way: the Repeating Label command, which tells the VisiCalc program to repeat a character (or series of characters) across the cursor location. Move the cursor to A6 and type **/-** (the Repeating Label command). The prompt line reads *Label: Repeating*, asking you what character (or characters) to repeat across the location.

Type an asterisk (\*) followed by **RETURN**. A6 now shows 12 asterisks. Type **/=** again, then **\*= RETURN**. Now A6 reads **\* = \* = \* = \* = \***, because the pair of characters you typed is repeated. You can type as many characters as you wish; the combination is repeated as many times as it fits in the column.

Now type / — — **RETURN**. A6 fills with hyphens, making a line across column A. A6 contains more than 12 hyphens, however. It will always be filled with hyphens, no matter how wide or narrow the columns. Try it. Change the width of column A to 20 by typing /GCC20 **RETURN**. Sure enough, A6 is still filled with hyphens. Type /GCC12 **RETURN** to return the width of column A to 12.

You can replicate a repeating label just as you can replicate a value. Type the following:

**/R RETURN → . N6 RETURN**

A line of dashes extends all across the worksheet. Scroll across to the totals (column N) if you want to verify it. Now draw a double line across the worksheet between the last line item and the monthly totals. Type the following:

**> A11 RETURN**  
**/ — = RETURN**  
**/R RETURN → . N11 RETURN**

This double line overwrites the unneeded formula at N11 that calculated the sum of a blank row. The screen should look like Figure 2-13.

The screenshot shows a Visicalc worksheet with the following data:

Salaries	Jan	Feb	Mar	Apr	May	Jun	Jul
Secretary	1000	1000	1000	1000	1000	1000	1000
Receptionist	800	800	800	800	800	800	800
Accounts Clerk	850	850	850	850	850	850	850
Office Mgr.	1250	1250	1250	1250	1250	1250	1250
	3900	3900	3900	3900	3900	3900	3900

The table is displayed on a screen with a grid. A double line is drawn across the row containing the 'Office Mgr.' entry, separating it from the 'Totals' row below. The 'Totals' row shows a sum of 3900 for each month.

010-011/P

Figure 2-13. Single and Double Lines on the Worksheet

You have created a fairly complex collection of numbers and labels with not very many keystrokes. To emphasize how few it took, now you'll add another, slightly larger section to the budget. Type the following, thinking what each entry will do and watching the results:

```

> A14 RETURN
Phys. Plant ↓
/ - - RETURN
/R RETURN → .N15 RETURN ↓
Rent ↓ Utilities ↓ Insurance ↓
Maintenance ↓ Furniture RETURN
> B16 RETURN
1300 ↓ 200 ↓ 400 ↓ 150 ↓ 500 ↓ ↓
@SUM(B16. ↑ ) RETURN
> B16 RETURN
/R.B22 RETURN → .M16 RETURN RR
> N16 RETURN
@SUM(B16.M16) RETURN
/R RETURN ↓ .N22 RETURN RR
> A21 RETURN
/ - = RETURN
/R RETURN → .N21 RETURN
    
```

You've more than doubled the size of the budget. Now you have 9 line items in two categories, 108 monthly figures with automatically updated vertical and horizontal totals. The screen should look like Figure 2-14.

Salaries	Jan	Feb	Mar	Apr	May	Jun	Jul
Secretary	1000	1000	1000	1000	1000	1000	1000
Receptionist	800	800	800	800	800	800	800
Records Clerk	650	650	650	650	650	650	650
Office Mgr	1250	1250	1250	1250	1250	1250	1250
	3900	3900	3900	3900	3900	3900	3900
Phys. Plant							
Rent	1300	1300	1300	1300	1300	1300	1300
Utilities	200	200	200	200	200	200	200
Insurance	400	400	400	400	400	400	400
Maintenance	150	150	150	150	150	150	150
Furniture	500	500	500	500	500	500	500
	2550	2550	2550	2550	2550	2550	2550

010-012/P

**Figure 2-14. Expanded Budget**

The final budget would have more entries, of course, but for the purposes of this lesson you have entered enough. A row of monthly grand totals completes this budget. Type the following:

```
> A23 RETURN  
/ - = RETURN  
/R RETURN → .N23 RETURN → ↓  
@SUM(B12, ↑ ↑ ) RETURN  
/R RETURN → . N24 RETURN RR
```

## **MORE ON THE SUM FUNCTION**

The Sum function at B24 specifies two locations (B12 and B22), rather than a range of locations like the earlier Sum functions. The locations are separated by a comma; a range of functions, remember, is separated by an ellipsis. The Sum function can take any number of arguments, separated by commas. Each argument can be a number, a location, a range of locations, another function, or a formula. The following are all valid Sum functions:

```
@Sum(B7,B8,B9)  
@Sum(B7...B12)  
@Sum(B7,B8,.03*B9)  
@Sum(B7,@Sum(F5...F10),B8 + 25 - C3)
```

Your budget automatically calculates monthly and yearly totals. It shows how quickly you can enter a VisiCalc worksheet that puts the power of the computer at your command.

## **CONTROLLING THE FORMAT OF ENTRIES**

The budget isn't as easy to read as it might be, because the column headings (Jan, Feb, Mar, etc.) don't line up with the numbers. The VisiCalc Format command lets you control how a label or value is displayed in the space available to it. Depending on the content of an entry, you can display it aligned with the left edge of the column, aligned with the right edge of the column, with two decimal places, with a dollar sign, a percent sign, or other qualities that make labels and values more readable.

## Setting a Local Format

Move the cursor to B5, and look at the column heading. Jan is aligned with the left edge of the column; this is how all labels are displayed unless you specify otherwise. But look at the numbers below the heading. They are aligned with the right edge of the column; this is how values are displayed unless you specify otherwise.

You can change this with the **Format** command. Type **/F**. The prompt line reads `Format: D G I L R $ * - = .` Each character following the colon (:) represents a different way of formatting data. **L**, for example, means display the value aligned with the left edge of the column; **R** means display it aligned with the right edge. This seems to offer a solution to the column-heading problem; type **R** to align Jan with the right edge of column B.

**NOTE:** The **Format** and **Attribute** commands offer many additional ways to control the format of an entry. You can review all the options now, or any time you wish, by turning to "Attribute Command" and "Format Command" in the Reference. Read as much or as little as you choose. The Reference can be an excellent aid to learning the VisiCalc program, and is the source for all details not covered in the lessons.

## Replicating a Format

Jan now lines up with the numbers beneath it, but it runs into Feb. Each monthly heading should be aligned with the right edge of its column. You could type **/FR** at each heading (C5 through M5), but the **Replicate** command lets you replicate just the format.

Type **/R**. The prompt line reads `Replicate: [, Source Range or RETURN.` For the first time you need to replicate a format. Type **(;** the prompt line reads `Replicate: Limit: A C N R ]`. Each character after **Limit:** represents an option of the **Replicate** command. The **C**, for instance, specifies that you want only the contents of a cell or cells to be replicated, not any formats or attributes that may be set in the cell.

The **A**, or **Attribute**, option specifies that you want only the format or attribute in a cell to be replicated, not the contents (data). Type **A**. The edit line reads `[A]`. Press **RETURN**. The edit line read `[A]B5`. Now only the format in B5 (**/FR**) will be replicated.

After you specify a limit or option, the **Replicate** command works as before. Press **RETURN** to enter the source range on the edit line; the edit line reads `[A]B5..B5`. Type **→.N5** to specify the target range. The edit line reads `[A]B5..B5:C5..N5`. Press **RETURN** to complete the command. The column labels line up with the numbers beneath them.

## CREATING A BUDGET CALCULATOR

One last change makes the budget even more useful. You have written each entry as a number. What if the rent goes up to 1500 in May? To make that change, you would have to enter 1500 in F16 through M16. As usual, the Replicate command offers an easier way to do this.

In the rent row, each entry contains the number 1300. The entries for February through December could just as easily consist of a reference to the rent for the preceding month. The appearance of the worksheet wouldn't change, but there would be an important difference: any change you entered would affect all remaining months because of the references to previous entries.

Try it. Move the cursor to C16; it reads 1300. Type **+B16 RETURN**. C16 still reads 1300, but move the cursor left to B16 and type **1500 RETURN**. Now both B16 and C16 read 1500. D16 still reads 1300 because the entry there is 1300. If you replicated the relative formula at C16 across the rent row, however, each location would depend on the previous entry; any value you enter affects all entries to the end of the year. Move the cursor to B16 and type **1300 RETURN**.

It takes only a few keystrokes to replicate this simple formula down and across the entire bottom half of the budget. Type the following:

```
>C16 RETURN
/R RETURN ↓ . ↓ ↓ ↓ ↓ RETURN R
/R ↓ ↓ ↓ ↓ RETURN → .M16 RETURN RRRRR
```

The upper half is almost as easy:

```
>C7 RETURN
+B7 RETURN
/R RETURN ↓ . ↓ ↓ ↓ ↓ RETURN R
/R ↓ ↓ ↓ ↓ RETURN → .M7 RETURN RRRR
```

Each entry in the budget — except for the first month — now refers to the previous month's entry. You can quickly forecast salary increases, reductions in utilities due to energy conservation, or other changes that continue until another change or the end of the year.

You have more than a budget; you have a budget calculator that quickly answers what-if questions. Increase the secretary's salary to \$1100 in June (type **>G7 RETURN 1100 RETURN**); the accounts clerk's position is being centralized in the district office, so change the salary to 0 in October (**>K9 RETURN 0 RETURN**); increase rent to 1500 in August (**>I16 RETURN 1500 RETURN**); and lower the utilities bill to 160 in July (**>H17 RETURN 160 RETURN**). With each change, the monthly and yearly totals are revised.

If proposing budgets is part of your responsibility, your productivity has just increased dramatically. You may find this budget form useful for later reference; save it by typing **/SS.D2/BUDGET.VC RETURN**.

**NOTE:** When you enter a number at a location, the number replaces the relative formula. If you enter a number then change your mind, remember to restore the entry to a reference to the previous month's entry. For example, if you increased the receptionist's salary to \$875 in April by entering 875 at E8, then decided to make the change in July instead, you would type not only 875 at H8, but also **+D8** at E8.

## PRINTING THE WORKSHEET

If you have used your printer before, the following instructions should let you print this budget with little difficulty. (If you haven't printed with your printer yet, you might want to wait until printing is described in more detail later.) The following instructions are for a Silentype printer. For a complete description of printers and how to print a worksheet, see "The Print Command" in the Reference and "Appendix B: Configurations for the VisiCalc(AV) Program."

To print the budget, move the cursor to A1 and type **/PP** (the P option of the Print command). These entries are described later. The prompt line reads *Print: Lower right* or "Setup. The VisiCalc program is waiting for you to specify the lower-right corner of the area you want to print.

You can print the entire worksheet by specifying N24 (the yearly grand total); you can print a smaller portion by specifying its lower-right corner. To print the first 6 months, including all line items and totals, make sure the printer is turned on and has paper, then type **G24 RETURN** (the grand total row for June).

The printer should print the first six months of the worksheet. As you will see later, if your printer is large enough to print on 15" paper, or if it can print smaller characters, you can print the entire worksheet at one time. If your printer can't print enough columns, it's still a fairly simple matter to print the worksheet in sections and tape them together.

**SUMMARY**

This lesson completes your introduction to the basic features of the VisiCalc program. You've learned how to write labels and values, how to change entries after you've written them, how formulas can relate different entries on a worksheet, and the key to building a worksheet: the Replicate command.

You're ready to strike out on your own. Although the VisiCalc program has many more capabilities, the commands and features you have learned are enough to build useful worksheets. There are many good applications similar to the branch office budget you just entered, such as a sales forecast or manufacturing schedule. Subsequent lessons will teach you more capabilities of the VisiCalc program, and techniques that let you build worksheets quickly and easily.

## LESSON THREE

Entering a Multi-Column Label	2-42
Formatting Left and Right	2-44
Formatting Integers and Dollars-and-Cents	2-46
Splitting the Screen	2-48
Replicating a Row	2-50
Summary	2-52

---

In the previous lesson, you used the VisiCalc program to build a budget. The VisiCalc program lends itself naturally to financial time-line applications, such as sales forecasts, amortization schedules, manufacturing schedules, cash flows, and many others. It can also be used for other types of applications that take advantage of recalculation, relative referencing, and other features.

This lesson uses an invoice to introduce many more features of the VisiCalc program. Figure 2-15 is a printed copy of the invoice you will enter. Refer back to it as you do the lesson to check your entries or to see how one part fits into the entire invoice.

Midwest Hardware Supply  
327 E. Mountain Avenue  
Berthoud, CO 80513

No: 123  
Date: 10/16/83

Bill to:  
-----  
Ranch Supply Co.  
2025 S. Main  
Longmont, CO 80310

Ship to:  
-----  
SAME

-----  
Terms: Net 30 Ship via:Truck Resale #: 1234567 Cust. #: 200  
=====

Item Number	Desc.	Quantity Ordered	Quantity Shipped	Retail Price	Disc. Price	Total Price	Back Ordered
20416	Gloves	20	10	5.60	4.15	41.50	10
20446	Glue	36	36	1.89	1.32	47.52	0
21329	Hammer	15	10	6.95	4.17	41.70	5
21377	Nails	50	50	1.90	1.43	71.50	0

-----  
Subtotal 202.22  
Tax 14.16  
-----  
Total 216.38  
Amt. Rcv.  
=====  
Amt. Due 216.38

010-004

**Figure 2-15. Invoice**

An invoice is a bill from a seller to a buyer for goods or services provided. The top part of an invoice includes the name of the seller, name of the buyer, invoice number, date, terms, and other information about the parties and terms of the transaction. The bottom part lists the items or services ordered, how many were delivered, price per item, total price for the quantity of each item, shipping charges, tax, and other information about the commodities of the transaction.

Many of the entries are repetitive, and some offer opportunity for human error. If the price of the items must be looked up, the wrong price can be written on the invoice, or an error can be made multiplying the price of an item by the number of items shipped. In either case, the invoice will bill the buyer for too much or too little. The VisiCalc program helps in both these areas.

## ENTERING A MULTI-COLUMN LABEL

The name and address of the selling company is usually centered at the top of the invoice. They are too wide to fit in one column, unless you make the columns quite wide; if you make them wide enough for the name and address, however, they will be much wider than needed for the numbers that appear below. There is a simple solution; type the title in segments—exactly one column wide—in adjacent columns.

Clear the worksheet (type **/CAY**). Move the cursor to D3 and type the following:

**Midwest H → ardware S → upply RETURN**

Although you entered the name in adjacent columns, it appears to be a single entry. You'll probably use this technique frequently in your worksheets. Remember, however, that if you later change the global column width — or the width of any individual column that contains a portion of the title — you'll have to reenter the title. To see the effect of this, change the global column width to 7 by typing **/GC7 RETURN** and look at the name. It's no longer readable. You would have to reenter it if you changed the column width permanently to 7.

Change the column width back to 9 (type **/GC9 RETURN**) and enter the rest of the invoice title by typing the following:

**>D4 RETURN**

**"327 E. Mo → untain Av → enue ↓ ← ←  
Berthoud, → " CO 80513 RETURN**

The quotation mark before 327 is needed to identify the entry as a label. The quotation mark before CO (the state abbreviation) is needed because a space is not recognized as the first character in a label. Type "; the prompt line reads *Label*. Now, each time you press the space bar, a blank is put in the label. Cancel this entry with **CONTROL-C**.

Enter the billing and shipping addresses by typing the following:

```
> A8 RETURN
Bill to: ↓
/— — RETURN
/R RETURN → . → → RETURN ↓
Ranch Sup → ply Co. ↓ ←
"2025 S. M → ain ↓ ←
Longmont, → " CO 80310 RETURN
> F8 RETURN
Ship to: ↓
/— — RETURN
/R RETURN → . → → RETURN ↓
SAME RETURN
```

The shipping addresses, like the titles, are entered in adjacent columns. Compare what you have entered to Figure 2-16. If there are any mistakes, erase them with the Blank command (/B) and retype them.

To complete the top part of the invoice, enter the date, invoice number, terms, and other details of the transaction by typing the following:

```
> G3 RETURN
No: → 123 ↓ ↓ ←
Date: → "10/16/83 RETURN
> A14 RETURN
/— — RETURN
/R RETURN → . H14 RETURN ↓
Terms: → Net 30 →
Ship via: → Truck →
Resale #: → 1234567 →
Cust. #: → 200 RETURN
> A16 RETURN
/— = RETURN
/R RETURN → . H16 RETURN
```

The top part of the invoice is now complete. The screen should look like Figure 2-16.

```

          Midwest Hardware Supply   No:      123
          327 E. Mountain Avenue
          Berthoud, CO 80513        Date:   10/16/83

Bill to: -----                Ship to: -----
Branch Supply Co.                SAME
2625 S. Main
Longmont, CO 80516

-----
Terms: Net 30  Ship via Truck  Resale #: 1234567  Cust. #: 800
-----

```

010-013/P

Figure 2-16. Shipping Addresses of Invoice

## FORMATTING LEFT AND RIGHT

As with the budget in Lesson One, the format of the entries needs to be adjusted a bit to make the invoice more readable. The labels and corresponding values for invoice number, date, resale number, and customer number are a bit too far apart. Type the following:

```

> G3 RETURN
/FR ↓ ↓ /FR → /FR
> F15 RETURN
/FL → → /FL

```

You have aligned the date at H5 with the right edge of the column to leave a space between Date: and the first character of the date (remember, the date is entered as a label, so it is aligned with the left edge of the column unless you specify otherwise).

The lower part of the invoice includes the column headings, item descriptions, quantities, price, shipping charges, and tax. Using the automatic recalculation feature of the VisiCalc command, the invoice lets you enter the item numbers, description, quantity, and price. It then calculates the total price and back order for each item, and the invoice totals.

Begin by entering the column headings for the body of the invoice. Type the following:

```
> A17 RETURN
Item → → Quantity → Quantity → Retail →
Disc. → Total → Back RETURN
> A18 RETURN
Number → Desc. → Ordered → Shipped →
Price → Price → Price → Ordered RETURN
> A19 RETURN
/— — RETURN
/R RETURN → . H19 RETURN
```

Compare your work to Figure 2-15.

To enter the first item, type the following:

```
↓ 20416 → Gloves → 20 → 20 → 5.60 → 3.64 RETURN
```

These are the six pieces of data you must enter for each item. Now you'll enter formulas so the VisiCalc program can calculate Total Price (Discount price multiplied by Quantity Shipped) and Back Ordered (Quantity Ordered less Quantity Shipped). Type the following:

```
→ + ← * ← ← ← RETURN
→ + ← ← ← ← ← — ← ← ← ← RETURN
```

Both formulas start with a plus sign to tell the VisiCalc program that you're entering a value. If you just typed the left arrow, the VisiCalc program wouldn't know that you want to enter a value and would obligingly move the cursor without starting a formula.

Now you have one complete row of items. At this point, you're entering more data than the VisiCalc program is calculating, but as you develop the worksheet the VisiCalc program will do more of the work for you.

## FORMATTING INTEGERS AND DOLLARS-AND-CENTS

Before entering any more items, see if there are any improvements you can make in the format of the worksheet. Look at the prices. Although you entered 5.60 for the retail price, it is displayed as 5.6; the calculated total price is 72.80, but it is displayed as 72.8. The general format for numbers (if you don't specify otherwise) is to display the number with the greatest precision possible in the available space. It displays 5.60 as 5.6 because the zero adds nothing to the precision.

You can force the VisiCalc program to display a number with exactly two decimal places by using the Dollars-and-Cents format. Move the cursor to E20 and type /F (the Format command). The prompt line reads Format: D G I L R \$ \* - = . You previously used the L (Left) and R (Right) options. The following list explains all the options.

**D Default** — This means display the location with the format last specified by a Global command. If no Global Format command has been issued, it means display the location with the General format.

**G General** — This is the format in which labels and values are displayed unless otherwise specified. Labels are aligned with the left edge of the column. Values are displayed aligned with the right edge of the column to the maximum precision possible in the space available.

**I Integer** — Displays a value as an integer. If the number has a decimal fraction, the number is rounded to the nearest integer. (Applies only to values.)

**L Left** — Aligns the entry with the left edge of the column.

**R Right** — Aligns the entry with the right edge of the column.

**\$ Dollars-and-Cents** — Displays a value with exactly two decimal places. Integers are displayed with a decimal point and two zeros. Values with more than two decimal places are rounded.

**\* Graph** — Displays a value as a string of asterisks equal to the value (1 is displayed as \*, 2 as \*\*, etc.). Values with decimal fractions are rounded to the nearest integer. If the column isn't wide enough to display the full value, the column is filled with asterisks (if the column width is 9, for example, all values greater than 9 are displayed as 9 asterisks).

– Repeating — Repeats a character or series of characters across a cell. Character(s) fill the cell regardless of column width.

= User-defined — Lets you assign a format name to a series of attributes. After assigning a one-character format name, typing /F and the format name will assign the series of attributes to a cell.

The \$ represents the Dollars-and-Cents format; type \$. Now E20 (the retail price) reads 5.60. You could format the other two prices (Disc. Price at F20 and Total Price at G20) by setting the global format to Dollars-and-Cents; type /GF\$.

Now the other prices are correct, but look at column H (Back Ordered) or column A (Part Number). They should be displayed as integers, not with two decimal places. You're going to have to set some local formats.

Return the global format to General by typing /GFG. Change the other two price formats to Dollars-and-Cents by moving the cursor to F20 and typing /F\$ → /F\$.

The numbers don't line up well with the headings (Item Number, Quantity Ordered, etc.), so use the Format command again to improve the appearance. Move the cursor to A20 and type the following:

**/FL → → /FL → /FL**

The prices still aren't lined up, but you can't align them left. The VisiCalc program displays Dollars-and-Cents values aligned with the right edge of the column. If you try to align them with the left edge of the column, the Dollars-and-Cents format is canceled, and the format reverts to General.

Try it. Move the cursor to E20 and type /FL. E20 reads 5.6, aligned with the left edge of the column. That's not the way numbers should look on an invoice, so return to the Dollars-and-Cents format by typing /F\$.

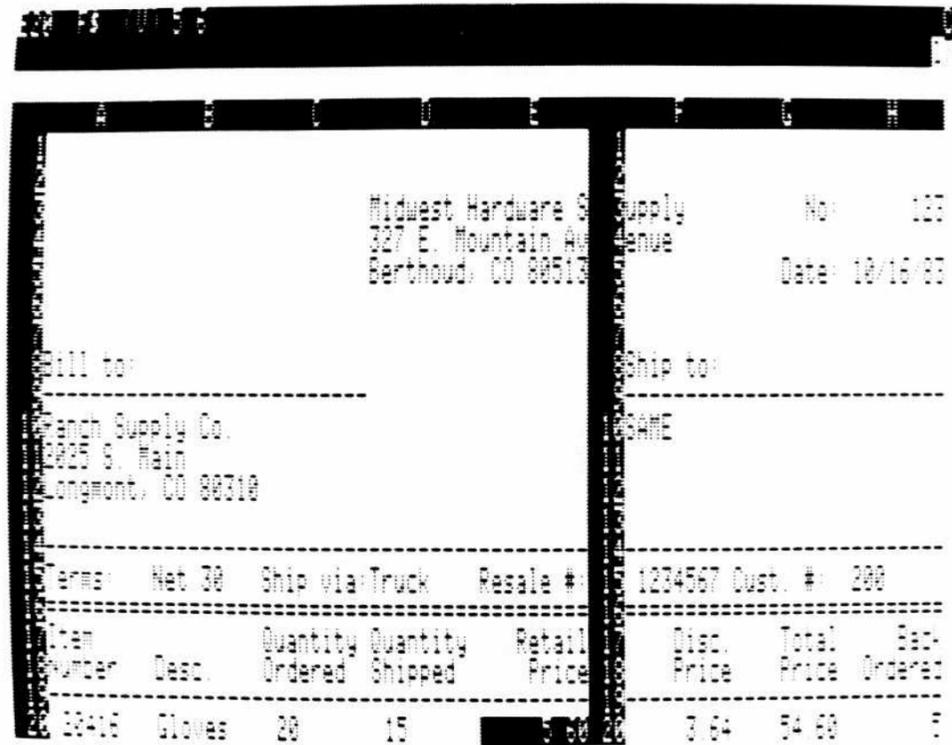
To correct the format of the price columns, move the headings to the right instead of the numbers to the left. Type the following:

**> E17 RETURN**  
**/FR → /FR → /FR → /FR**  
**> E18 RETURN**  
**/FR → /FR → /FR → /FR**

You moved the last heading (Back Ordered) to the right to keep it separated from the preceding heading (Total Price).



The screen is divided by a white bar between columns E and F. The row numbers are repeated here, so that you can keep track of the cursor in either window as you scroll. The cursor is at E20 in the left window. The screen should look like Figure 2-18.



010-015/P

Figure 2-18. Screen Split Vertically

Scroll to the left until the cursor reaches the left edge of the window; the right window doesn't move.

The semicolon (;) moves the cursor to the other window. Type ;. The cursor moves to F20 in the right window. Again, scroll left until the cursor reaches the left edge of the window. Only the right window moves.

Type ;. The cursor moves back to A20 in the left window. Type ; again. The cursor moves to A20 in the right window. When the cursor moves to the opposite window, it returns to the last position it occupied.

One way the split window can be useful is to see two distant parts of the worksheet at one time. Scroll the right window until columns G and H (Total Price and Back Ordered) are visible. Move the cursor back to the left window, then to F20. Type **4.15 RETURN** to change Discount Price, and watch Total Price change in the right window. Move the cursor to D20, and type **10 RETURN** to change Quantity Shipped. Total Price changes to 41.50 and, if you haven't changed Quantity Ordered, Back Ordered changes to 10.

Change back to one window by typing **/W1**.

**REPLICATING A ROW**

Enter three more items on the invoice by typing the following:

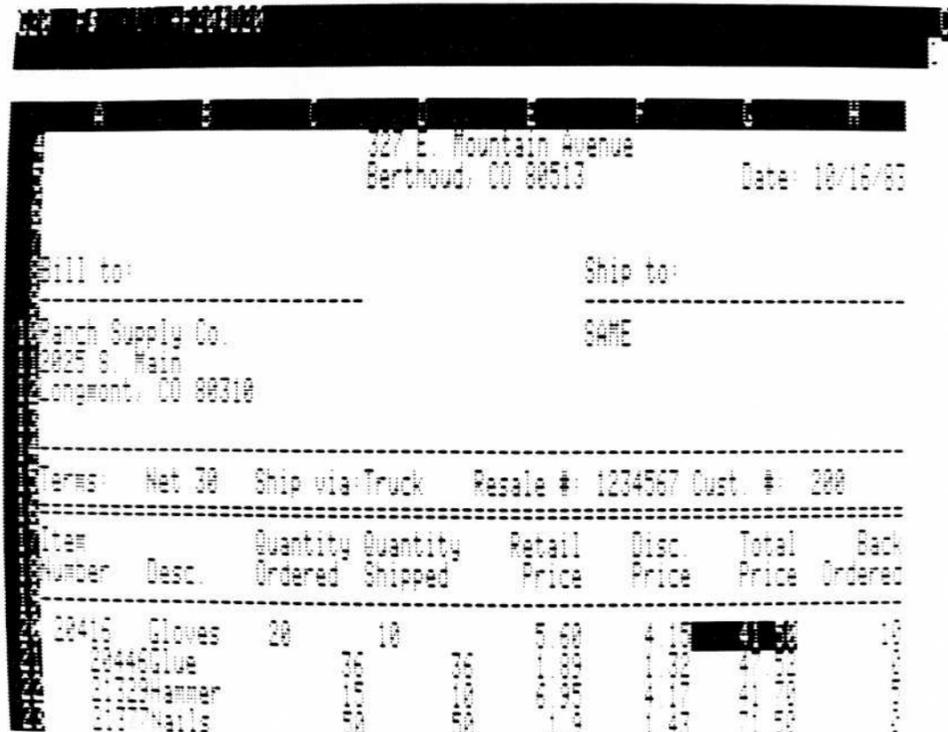
- > **A21 RETURN**
- 20446 → Glue → 36 → 36 → 1.89 → 1.32 RETURN**
- > **A22 RETURN**
- 21329 → Hammer → 15 → 10 → 6.95 → 4.17 RETURN**
- > **A23 RETURN**
- 21377 → Nails → 50 → 50 → 1.90 → 1.43 RETURN**

The VisiCalc program can't calculate Total Price or Back Ordered because there are no formulas on the worksheet in their rows. The calculations are the same as for the first item, so you can replicate the formulas you entered in the first row to take care of these additional rows. Type the following:

- > **G20 RETURN**
- /R → RETURN ↓ . ↓ ↓ ↓ RETURN RRRR**

The VisiCalc program asked you to specify No change or Relative four times: once each for the two location references in the formula for Total Price, and once each for the two location references in the formula for Back Order.

You now have a fairly complete invoice. The screen should look like Figure 2-19.



010-016/P

**Figure 2-19. Invoice With Several Items**

The Quantity Ordered, Quantity Shipped, and Retail Price entries aren't formatted properly for these new items. Replicate the format from the first row by typing the following:

```
> A20 RETURN
/R(A RETURN
.→ →→→→RETURN
↓.↓↓↓ RETURN
```

The invoice needs a totals block at the bottom. It includes subtotal, sales tax, total, amount received, and amount due. Type the following:

```
> F29 RETURN
Subtotal ↓ Tax ↓↓
Total ↓ Amt. Rcv. ↓↓ Amt. Due RETURN
> G29 RETURN
/F$@SUM(G20. ↑ ) ↓
/F$.07 * ↑ RETURN ↓↓
/F$@SUM( ↑↑↑. ↑ ) ↓↓↓
/F$+ ↑↑↑-↑↑ RETURN
> A28 RETURN
/— — RETURN
/R RETURN → .H28 RETURN
> F31 RETURN
/— — → /— — RETURN
> F34 RETURN
/— = → /— = RETURN
```

The screen should look like Figure 2-20.

Item Number	Desc.	Quantity Ordered	Quantity Shipped	Retail Price	Disc. Price	Total Price	Back Ordered
20416	Gloves	20	10	5.60	4.15	41.50	10
20446	Blue	36	36	1.03	1.03	47.52	0
21708	Hammer	10	10	6.95	4.17	41.70	0
21377	Nails	50	50	1.90	1.43	71.50	0
						Subtotal	202.22
						Tax	14.16
						Total	216.38
						Amt. Rcv.	
						Amt. Due	216.38

010-017/P

**Figure 2-20. Totals Block of Invoice**

Save this version of the invoice by typing **/SSINVOICE.VC RETURN**.

## **SUMMARY**

The invoice you entered in this lesson shows how the VisiCalc program can be used for applications that don't depend on a series of periodic entries (a time line). Techniques like splitting the screen and specifying formats for entries make your worksheets more useful.

You can apply these techniques to similar applications, such as an inventory report or tax calculator. As you add more VisiCalc capabilities to your repertoire, the range of possible applications expands.



If you're starting a new session, load the worksheet named INVOICE.VC you saved in the previous lesson. If you're continuing from the previous lesson and you didn't save the invoice, please go back and do so (see "Saving the Invoice" at the end of the lesson).

## DISPLAYING FILE NAMES

To begin this lesson, clear the worksheet (/CAY) and type the Load option of the Save command: /SL. Instead of typing the file name, however, type →. A file name appears on the edit line. Type → again. If you have stored more than one file on the diskette, another file name appears on the edit line.

Display the file names until the name you gave the invoice is on the edit line; then press RETURN. This is exactly the same as typing the file name and RETURN. The prompt line reads Storage: Loading followed by the blinking asterisk (\*) that tells you a worksheet is being loaded.

The invoice you saved is ready to be used as it is. You enter the customer and transaction data, then the name, quantity, and price information for each item. The VisiCalc program calculates the total price for each item, sales tax, and invoice total, resulting in an invoice ready to be printed.

But the VisiCalc program can do more for you. In this lesson, you will learn how to let the program select the largest of several values, select a value depending on a condition, look up item numbers and prices, calculate discounts, etc.

## INSERTING COLUMNS AND ROWS

Sometimes you'll want to add something in the middle of a worksheet. It's not necessary to reenter the worksheet; you can insert rows or columns, then write the new entries in these blank rows or columns. (You can also, of course, insert a row or column simply to add blank space to the worksheet).

Suppose you decide to add shipping charges to the totals block of the invoice, just below Subtotal. Everything below Subtotal must move down one row. But the formula for Total (at G32) refers to Tax (at G30); references to G30 must be changed if the location of Tax changes. The VisiCalc program adjusts all references to locations when you insert rows or columns.

Try it. Insert a new row just below Subtotal. But first, move the cursor to G32 and look at the entry line. It reads G32 /FS (V) @SUM[G29...G31], defining the total as the sum of Subtotal and Tax. Now move the cursor to G30 and type /I (the Insert command). The prompt line reads Insert: R C or Range. You want to insert a row, so type R. A blank row appears at row 30 and all rows below it move down to make room. Move the cursor to F30 and type **Shipping RETURN**.

Check the formula for Total (now at G33) again; the entry line reads G33 /FS (V) @SUM[G29...G32]. The range of the Sum function has expanded to accommodate the new row for Shipping. Similarly, the formula for Amt. Due at G36 has changed from +G32-G33 to +G33-G34.

## DELETING COLUMNS AND ROWS

You can also delete columns and rows. Again, the VisiCalc program corrects all references to cells that move as a result of the deletion. Be sure the cursor is still in row 30 (the row you just inserted) and type /D. The prompt line reads Delete: R C or Range. Type R.

Now the prompt line reads Delete: Type Y To Confirm; the program requires you to confirm the deletion, minimizing the chance that you'll inadvertently delete an entry. Type Y to confirm the deletion, and the row is gone. References to rows below the deletion (which moved up to fill the gap left when row 30 was deleted) have been adjusted.

**WARNING:** If you delete a row that contains cells referenced elsewhere on the worksheet, all such references are changed to Error because the referenced cells no longer exist.

With the cursor at F30, type /IRShipping RETURN to insert the row for shipping charges again.

## SELECTING THE LARGEST OF SEVERAL ENTRIES

Now that you have room for shipping charges, what should the entry be? It's fairly common for shipping charges to be calculated as a percentage of the charge for all items, but with some minimum charge to cover small orders. Stated another way, it is either a percentage of the total or some specific amount, whichever is greater: the *maximum* of those two values. The VisiCalc Maximum function makes that choice.

The Maximum function (entered `@MAX`) takes any number of arguments (a list), separated by commas. The value of the function is the value of the largest argument. Figure 2-22 shows several Maximum functions and their values.

	A	B	C	D	E	F	G	H
1								
2								
3	10	5	60	.25	100			
4								
5								
6								

The value of `@MAX (A3, B3)` is 10  
 The value of `@MAX (A3...E3)` is 100  
 The value of `@MAX (A3...C3, D3*E3)` is 60  
 The value of `@MAX (D3*E3, 20)` is 25

010-006

**Figure 2-22. Examples of Maximum Function**

To set shipping charges to 5 percent of the invoice subtotal or \$15, whichever is greater, move the cursor to G30 and type the following:

`/F$@MAX(.05* ↑ ,15) RETURN`

Shipping is 15, because 5 percent of Subtotal is less than 15. Increase Subtotal by ordering 100 hammers: move the cursor to C22 and type `100` → `100 RETURN`. Now move the cursor back to G30; Shipping is 28.88, because 5 percent of Subtotal (577.52) is more than 15.

## SELECTING A VALUE THAT MEETS A CONDITION

Sometimes you want the value of one location to depend on a condition involving the value of other locations. Consider sales tax, for example. The invoice should not add sales tax if the items are being purchased for resale. One of the items in the top half of the invoice is Resale #, a license issued by state government authorizing an individual or firm to sell products and collect sales tax. *If* a value is entered for Resale Number (at F15), the buyer should not pay sales tax.

The If function makes such a choice. It requires 3 arguments: the first argument describes a *condition* that is either true or false, the second is the value of the function if the condition is true, and the third is the value of the function if the condition is false.

### **Describing a Comparison Condition**

A condition statement usually consists of two values separated by a *comparison operator*; it is either true or false. The table below shows some condition statements.

<b>Comparison Operator</b>	<b>Meaning</b>	<b>Example</b>
=	Equal to	B1 = C1
>	Greater than	F10 > 0
<	Less than	A3 + A4 < C3
> =	Greater than or equal to	D9 > = E9
< =	Less than or equal to	B3 - 7 < = 1
< >	Not equal to	C5 < > (C1 + C2)

**NOTE:** If the value on the right side of the comparison operator is more than a single element, it must be enclosed in parentheses, as is C1 + C2 in the last condition statement.

The value of a condition statement is one of the special VisiCalc values @True or @False (displayed as True or False). For example, 5 = 3 is obviously False; 125 > 7 is obviously True. But the value of B1 > B2 depends on the values of B1 and B2 at the time the condition statement is evaluated; it can change as you change values on the worksheet.

Like many VisiCalc features, it's easier to understand the If function when you see it. Figure 2-23 shows a portion of a worksheet with several If functions.

	A	B	C	D	E	F	G	H
1								
2								
3			25	25	50			
4								
5	10							
6	20							
7								
8								

The value of @IF (A5 > A6, D3, E3) is	50
The value of @IF (A5 + A6 > @SUM (C3...E3), .5 * A6, 0) is	0
The value of @IF (D3 - 15 > = A5, A5 * D3, A5 * E3) is	250
The value of @IF (D3 = E3, 0, @IF (D3 > E3, D3 - E3, E3 - D3)) is	25

010-007

Figure 2-23. If Function Examples

This description of the If function all began with a discussion of when and how to charge sales tax. Following the general form of the VisiCalc If statement, the choice is described by the following words:

If the customer has a resale number, there is no sales tax (sales tax is 0), otherwise sales tax is Subtotal (G29) multiplied by the tax rate.

Move the cursor to G31 (the formula for Tax), and type the If function that corresponds to the verbal description of the choice:

```
@If(F15 > 0,
0,
↑↑ * .07) RETURN
```

Compare this with the verbal description; both say the same thing, line-for-line.

Now the tax entry depends on the resale number; if there is a resale number (F15 is greater than 0), tax is not added; otherwise tax is added. Right now, Tax is .00 because F15 is 1234567. Move the cursor to F15 and blank the entry. Tax (G31) changes to 40.43, and Total and Amt. Due change to 646.82. Change F15 back to 1234567; Tax, Total, and Amt. Due return to their former values.

Other VisiCalc functions let you select alternative values for an entry. The Choose function, for example, lets you list several values in the arguments; the first argument, called the *index*, specifies which of the values listed in the subsequent arguments is to be assigned to the function. For a full description of these functions, see "Functions" in the Reference.

## **INSERTING MULTIPLE COLUMNS**

To make room for the new material to be added in this lesson, you need four new columns left of the invoice. Move the cursor to **A1**, and type **/I** (the Insert command). The prompt line reads *Insert: R C or Range*. You can insert more than one row or column (a Range) by typing the number of rows or columns to be inserted, then either **R** for row or **C** for column.

Type **4C**. Four columns are inserted, moving the invoice over so it now begins in column **E**. (Inserting multiple rows works the same way.)

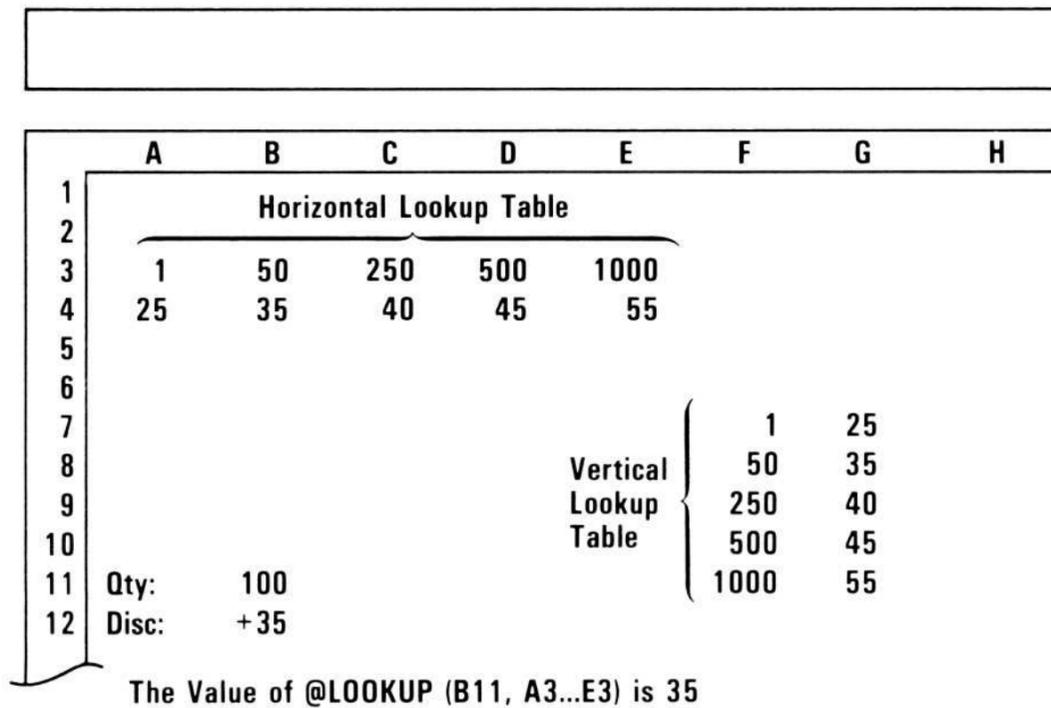
The Delete command also lets you type a number before the **R** or **C** to delete more than one column or row.

## **LOOKING UP VALUES IN A TABLE**

Although you have improved the invoice by letting it do routine chores, you can make it do even more. You still must look up item names, numbers, prices, and discount prices. You can make the invoice easier to use and less susceptible to error by adding a table that relates prices to the item number. The Lookup function does this.

The Lookup function requires two arguments: a *search value*, which is the value the function looks for, and the coordinates of the *search list*, a series of values in successive locations. A second series of values parallel to the search list contains the value assumed by the Lookup function when it finds the search value in the search list. The search list and corresponding values must be in adjacent columns or adjacent rows. The corresponding values must be below or to the right of the search list.

This is much less complicated than it sounds. Figure 2-24 shows both horizontal and vertical tables that can be used to look up discount rates. The search lists represent quantity purchased, ranging from 1 to 1000; the corresponding discount rates range from 25 to 55. The horizontal search list is **A3** through **E3**; the corresponding values are **A4** through **E4**. The vertical search list is **F7** through **F11**; the corresponding values are **G7** through **G11**. The Lookup function is at **B12**, and the quantity purchased is at **B11**.



010-008

Figure 2-24. Horizontal and Vertical Lookup Tables

The Lookup function compares the search value (the first argument) to each successive value in the search list (the second argument). It finds the largest value in the search list that is equal to or less than the search argument, and selects the corresponding value from the adjacent series.

For example, the Lookup function at B12 in Figure 2-24 specifies B11 as the search value and A3 through E3 as the search list. The list of percentages, therefore, is assumed to be in A4 through E4. The value of B11 is 100, so the largest value in the search list that is equal to or less than the search argument is 50, at B3; the corresponding percentage is 35 (at B4), so the value of the Lookup function is 35 (shown at B12). If B11 is changed to 400, the value of the Lookup function changes to 40; if B11 is changed to 500, the value of the Lookup function changes to 45.

The values of the Lookup function in Figure 2-24 would be the same if the vertical search list (F7...F11) were specified rather than the horizontal search list (A3...E3).

If the search value is less than the first value in the search list, the Lookup function evaluates to NA. This is a special value that means Not Available; the first value in the search list is greater than the search argument, so there is no corresponding value to assume. If the search value is greater than the highest number in the search list, the Lookup function evaluates to the last number in the list of corresponding values.

**WARNING:** For the Lookup function to operate properly, the values in the search list must be entered in ascending order.

**COPYING COLUMNS WITH THE REPLICATE COMMAND**

To use a lookup table for prices, you must enter the prices in a row or column adjacent to a list of the item numbers. Rather than reenter this data, you can copy it to the upper-left part of the worksheet with the Replicate command. Type the following:

**> E20 RETURN**  
**/R ↓ ↓ ↓ RETURN A4 RETURN**

The item numbers appear both on the invoice and in the upper-left part of the worksheet. Add the retail prices and headings for the lookup table by typing the following:

**> B4 RETURN**  
**5.9 ↓ 3.35 ↓ 7.75 ↓ .85 RETURN**  
**> A1 RETURN**  
**Item → Retail ↓ Price ← Number ↓**  
**/— — → /— — ↓**  
**/F\$**  
**/R(A RETURN RETURN ↓.↓↓↓ RETURN**

The screen should like Figure 2-25.



010-018/P

**Figure 2-25. Lookup Table for Retail Price**

With the lookup table in place, you need a formula in the Retail Price column to look up the prices. The search value is the item number, and the search list is B4 through B7. Move the cursor to the retail price for the first item (I20), and type the following:

**@Lookup( ← ← ← ← ,A4.A7) RETURN**

The retail price for gloves appears on the worksheet; the Lookup function has searched the list of item numbers in the upper-left part of the worksheet looking for the largest value equal to or less than 20416, the item number for gloves. The first value in the search list — 20416 — is equal to 20416, so the Lookup function evaluates to 5.90, the corresponding value from the adjacent list of prices.

Replicate the formula down the column of retail prices:

**/R RETURN ↓ . ↓ ↓ ↓ RETURN**

As usual, the VisiCalc program asks you how to replicate the references. You want the Lookup function to search for each item's part number, so that reference is Relative; the search table is the same for each part, however, so the beginning and ending references for the search table are replicated with No change. Type **RNN** to complete the command; the price of each item appears in the retail price column.

But you still have to look up the discount price. The worksheet can take care of this, too.

Discounts usually increase as the quantity purchased increases. You can add a lookup table for percent discount, similar to the examples in Figure 2-25. It consists of a list of quantities and a corresponding list of discount rates. Type the following:

**> C1 RETURN**

**Quantity → Percent ← ↓**

**/— — → /— — ← ↓**

**0 ↓ 1 ↓ 10 ↓ 25 ↓ 50 ↓ 100 ↓ 500 ↓ 1000 RETURN**

**> D3 RETURN**

**/FL0 ↓ /FL20 ↓ /FL+ ↑ +5 RETURN**

**/R RETURN ↓ . ↓ ↓ ↓ ↓ ↓ RETURN**

**R**

This is the lookup table for percent discount. The screen should look like Figure 2-26.



010-019/P

Figure 2-26. Lookup Table for Discount Percent

This corresponds to the discount schedule shown below. The table on the worksheet has entries for 0 (at C3 and D3) so that a Quantity Ordered of zero results in a discount of 0 rather than a value of @NA (Not Available).

Quantity	Percent Discount
1-9	20
10-24	25
25-49	30
50-99	35
100-499	40
500-999	45
1000 +	50

The percent discount is calculated with a Lookup function that specifies Quantity Ordered as the search value and C3 through C10 as the search list. Discount Price can be calculated by a formula that reduces the retail price by the percent discount. Type the following:

**>J20 RETURN**

**+ ← - ( ← \* (@Lookup( ← ← ← ,C3.C10)/100))**

**RETURN**

The formula says that Discount Price is Retail Price (I20) minus Retail Price multiplied by the percent discount (the value of the Lookup function divided by 100). Discount Price for 20 pair of gloves is 4.43. Move the cursor to G20, and change Quantity Ordered to **100**; Discount Price changes to 3.54, because the discount for 100 pair of gloves is 40 percent.

Move the cursor back to J20, and replicate the formula for Discount Price down the column:

**/R RETURN**

**↓ . ↓ ↓ ↓ ↓**

**RETURN**

The formula has five references to other locations that you must specify as Relative or No change to the Replicate command. Think about the references for a moment. The first two both refer to the retail price of the item and the third refers to the quantity ordered; these should all be relative. The last two are the coordinates of the search list for the Lookup function; the same table is used for each item, so these should not be changed. Type **RRRNN**.

The invoice is now an automatic entry form. You enter Quantity Ordered and Quantity Shipped; the worksheet calculates Discount Price, Total Price, and Back Ordered.

## MAKING A WORKSHEET MORE FLEXIBLE

The formula for Shipping at K30 specifies exact numbers for percentage of Subtotal (.05) and minimum amount (15). Similarly, the formula for Tax (at K31) specifies the exact percentage (.07). To change any of these values (to increase the minimum shipping charge to \$20, for example), you must reenter the formulas. If you write these values as separate entries, however, and refer to their locations in the formulas, you can change the values without reentering the formulas.

Type the following:

```
> A12 RETURN
/--- → /--- ← ↓
Tax and S → hipping ← ↓
/--- → /--- ← ↓
Sales tax ↓ Ship % ↓ Min. Ship ↑ ↑ →
.07 ↓ .05 ↓ 15 RETURN
```

These are the same values specified in the formulas for Shipping and Tax: the tax rate is 7 percent, shipping charge is 5 percent of the sub-total, and the minimum shipping charge is 15. Now you can change the formulas to refer to these locations.

## CHANGING AN ENTRY ON THE WORKSHEET

You could reenter the formulas for shipping charges and sales tax, but the VisiCalc program lets you change, or *edit*, an existing entry without retyping it. Long labels or values are much easier to edit than to reenter.

The Edit command puts the entry highlighted by the cursor on the edit line, just as if you had typed it; the edit cue (the small white box) is at the beginning of the line. You can move the edit cue with the arrow keys. Whatever you type is inserted to the left of the edit cue; whenever you type ESCAPE, the character to the left of the edit cue is erased.

Try it. Move the cursor to K30 (the formula for Shipping). It reads @MAX(.05\*K29,15). Type /E (the Edit command). The formula appears on the edit line; the edit cue is at the beginning of the line.

You want to replace .05 with a reference to the location that contains shipping percentage (B16). All edit actions take place just to the left of the edit cue, so move it until it highlights the asterisk. Press ESCAPE, and the 5 disappears. Press ESCAPE ESCAPE, and the left parenthesis is just to the left of the edit cue; you have erased .05. Type B16.

Now the edit line reads @MAX[B16\*K29,15]. You want to change the 15 at the end of the formula to B17 (the location that contains minimum shipping charge). Move the edit cue to the closing parenthesis and press ESCAPE ESCAPE B17 RETURN.

The formula for Tax at K31 reads @IF[J15 > 0,0,K29\*.07]. To change it to refer to the tax rate at B15, type ↓/E, move the edit cue to highlight the asterisk, and press ESCAPE ESCAPE ESCAPE B15 RETURN.

This completes the additions to the invoice for this lesson. It has become quite a bit more useful. You need enter only Quantity Ordered and Quantity Shipped for each item, plus any amount received with the order; the worksheet looks up Retail Price and percent discount, calculates Total Price and Back Ordered for each item, then Subtotal, Shipping, Tax, Total, and Amt. Due for the invoice.

## **EDITING A FILE NAME**

You will use this invoice in later lessons, so save it. Type **/SS** → to display the first file name on the diskette. Continue to type → until Invoice.VC is on the edit line.

You can edit this file name just as if you had typed it. Because this is a later version of the invoice you saved at the end of Lesson 3, you'll want to identify it as the second version of the invoice. The edit line now reads Invoice.VC. Type **ESCAPE ESCAPE ESCAPE2.VC** to identify this file as a VisiCalc worksheet; then press **RETURN** to save the worksheet.

## **SUMMARY**

The more advanced capabilities you learned in this lesson let you build more what-if capabilities into your worksheets. Features such as the Maximum, If, and Lookup functions let you delegate routine, repetitive tasks to the computer.

Now you can build a model of your business, and exercise the model to test alternative strategies. The power of the VisiCalc program is available to you to make the future less uncertain.

## LESSON FIVE

Hiding an Entry	2-68
Setting Tab Positions	2-69
Controlling the Type of Data That Can Be Entered	2-71
Protecting Cells Against Change	2-72
Displaying the Formulas of the Worksheet	2-72
Summary	2-74

---

Unlike a budget worksheet that might be used once or twice during a planning period, the invoice you created in Lessons 3 and 4 might be used several times a day. It has fixed entries that change infrequently (labels, prices, and part numbers) and variable entries that change each time the invoice is used (quantity ordered and quantity shipped).

The VisiCalc program gives you control over how an entry is displayed, what sort of data it can contain (label or value), whether it can be changed, even whether it is displayed. These characteristics are controlled by the **Attribute** command; this lesson shows you how to use the **Attribute** command to turn the invoice into a data-entry application to be used by someone else.

Start by clearing the worksheet and loading the file you saved at the end of the previous lesson (it was suggested you save it as Invoice2.VC). If you are starting this lesson without having completed Lesson Four, load the file named Invoice2.VC from your VisiCalc program diskette.

## HIDING AN ENTRY

Sometimes you need a worksheet entry to do a calculation but the person using the worksheet doesn't need to see the result of the calculation. The tables in the upper-left part of the invoice worksheet, for example, are used by the VisiCalc program to look up the retail price and discount, but aren't part of the invoice itself. If the invoice worksheet is used by someone else, it might be less confusing if the tables weren't displayed.

The H (Hide) option of the **A**tttribute command makes an entry invisible. Move the cursor to **A1** and type **/A**. The prompt line reads **Attribute: EHMTLVD**. The letters **EHMTLVD** are options; they are described in the following list.

- E** Expression — displays all values as expressions (the numbers or formulas entered, not the calculated result).
- H** Hide — makes the entry of this cell invisible.
- M** Mode — specifies what type of data (label, value, or number only) can be entered in this cell.
- T** Tab — defines this location as a tab stop.
- L** Label — specifies how labels are to be displayed in this cell.
- V** Value — specifies how values are to be displayed in this cell.
- D** Default — reverts attribute settings to global defaults.

Several of these options have further options of their own; attributes can be set locally (for one location only) or globally (for the entire worksheet). Some options are described in this lesson, others are described in later lessons. For a complete description, see "The Attribute Command" in the Reference.

Type **H**. The prompt line reads **Hide: Y N D !** and the edit line reads **HD:!** followed by the edit cue.

The **Y** means Yes, **N** means No, and **D** means Default. You want Yes, so type **Y**. The edit line changes to **HY:!**. That's how you specify Attribute command options. The colon and exclamation point (:) mark the end of the currently specified attributes; when you type a choice, the options to the left of the colon change. The attributes aren't actually assigned until you type RETURN; if you cancel the Attribute command with CONTROL-C, no change is made.

The **!** on the edit line puts attributes into effect in the cell as you specify them. This lets you see the effect of different combinations of attribute settings without entering your selections in memory. If you press **RETURN** after setting some attributes, your settings are entered into memory.

If you type **!** while the **!** is displayed on the edit line, attribute settings will not go into effect until you press **RETURN**, and the **!** will disappear from the edit line. This is useful when setting global attributes, since it may take several seconds to re-format the worksheet each time you make an attribute setting.

Press **RETURN**. The entry line reads `A1 /A HY!`, showing the attributes assigned to the cell. The prompt line reads `Attribute: E H M T L V D`, allowing you to select another attribute. Press **RETURN** again, and the promptline clears. Now the entry line reads `A1 /A (L) Item`, telling you that attributes are assigned to the cell highlighted by the cursor and showing its contents (Item).

Replicate the Hide attribute across the top row of the lookup tables by typing `/R(A RETURN RETURN → . → → → RETURN`. Now replicate this row down the lookup tables by typing `/R(A RETURN → → →: ↓ . A12 RETURN`. The lookup tables are no longer visible.

You change an entry that is hidden the same way you change any other entry; move the cursor to it and either edit or reenter the entry. The entry won't be displayed, however, unless you turn off the Hide attribute by typing `/AHN RETURN RETURN`.

**NOTE:** If you hide an area of a worksheet, first save a copy of the worksheet with all entries visible. Then, hide the entries you don't want seen, and save this version with a different name. This way, you lessen the danger of forgetting where a particular entry is and searching around to find it.

## SETTING TAB POSITIONS

Only certain entries on the invoice must be filled in. Rather than explaining to someone which entries must be filled out (or trying to remember yourself), you can direct the movement of the cursor to certain locations. This is similar to the use of the tab key on a typewriter.

In addition to RETURN and the arrow keys, you can move the cursor with the TAB key and back-tab key (SHIFT-TAB or CONTROL-B). Either key ends an entry and moves the cursor to the next location with the T (for Tab) attribute. The VisiCalc program searches across rows for the next Tab location. If there is no entry with the Tab attribute, the cursor doesn't move.

To see how this works, move the cursor to L3 (the invoice number), the first location to be filled in. Type **/ATY RETURN RETURN**. The next location to be filled in is L5 (Date), down two rows. Move the cursor to L5, type **/ATY RETURN RETURN**, and move the cursor back to L3. You have assigned the Tab attribute to L5.

Now type **201 TAB**; 201 is written at L3 as the invoice number and the cursor moves to L5, ready for you to enter the date. If you assign the Tab attribute to each entry that must be filled in, anyone filling in the invoice can move quickly through the invoice by ending each entry with TAB.

To assign the Tab attribute to each location to be filled in, type the following:

```
>E10 RETURN
/ATY RETURN RETURN
→/ATY RETURN RETURN
>J10 RETURN
/ATY RETURN RETURN
>F15 RETURN
/ATY → → /ATY → → /ATY → → /ATY RETURN RETURN
>G20 RETURN
/ATY → /ATY ←
/R(A→ RETURN ↓ . G23 RETURN
```

Now you can move quickly to the next tab location by pressing TAB or to the previous tab location by pressing CONTROL-B or SHIFT-TAB (the TAB and shift keys simultaneously). When you press TAB at the last tab location on the worksheet, the cursor moves to the first tab location; when you press CONTROL-B or SHIFT-TAB at the first tab location, the cursor moves to the last tab location.

## CONTROLLING THE TYPE OF DATA THAT CAN BE ENTERED

In addition to defining a location as a tab stop, you can specify the type of data that can be entered at a location. The M (Mode) option of the Attribute command lets you specify whether an entry can be a label, a value, or a number.

Move the cursor to L3, and type **/AM**. The prompt line reads `Mode: P U A L V # D |`, and the edit line reads `MD:|`. Each character of the Mode option is described in the following table. (The | does not set an attribute; it lets you show the effect of the attributes as you set them when a | is on the edit line, or it lets you not show the effect of the attribute when the | is not on the edit line.)

- P Protect (allows no changes).
- U Unprotect (allows changes).
- A All types (allows labels, values and numbers).
- L Label (allows labels only).
- V Value (allows values only).
- # Number only (allows numbers only).
- D Default (the global setting, if any).

A number should always be entered at L3, so type **# RETURN RETURN**. Now try to type a letter. The computer beeps an error. Try to type a formula. It won't let you. All you can type is a number.

The cells to be filled in include both numbers and labels: the invoice number, resale number, customer number, quantity ordered, and quantity shipped are all numbers; the date, customer name and address, and terms are labels. You have already specified the type of entry for the invoice number; type the following to specify the appropriate type at the remaining locations:

```
TAB
/AML TAB
/AML TAB /AML TAB
/AML TAB /AML TAB /AM# TAB /AM# TAB
/AM# → /AM# ←
/R(A → RETURN ↓. ↓ ↓ ↓ RETURN
```

The locations to be filled in will now accept only the type of data you have specified.

## PROTECTING CELLS AGAINST CHANGE

You have created a data-entry form that guides the user through it and accepts only the correct type of data at each cell. You can also protect all other cells against change. Again, you use the Mode option of the Attribute command. Start, however, with the Global command — because you want to set an attribute for all cells on the worksheet that don't have a local attribute — then select the Attribute option.

Type **/G** (the Global command), then **AM**. As before, the prompt line reads `Mode: P U A L V # D`. The **P** means Protect, and prevents anyone from changing the entry. The **U** means Unprotect; if a cell has been protected, you must use this option before you can change it. You want to protect the worksheet, so type **P RETURN RETURN**.

Now move the cursor to any location other than one to be filled in and try to type an entry. You can't. You have protected the entire worksheet; only those locations to be filled in can be changed. The Global Protect option doesn't affect the locations to be filled in because they are controlled by local attributes. Local attributes take precedence over Global attributes.

The invoice worksheet is ready for someone else to use. You need teach them only how to start the VisiCalc program, load this worksheet, fill out the locations, and save or print an invoice. You can have them fill out the worksheet with confidence that the worksheet itself will guide them through the entry procedure, provide some protection against incorrect data entry, and won't be damaged through inadvertant errors.

Again, save this version of the worksheet. Type **/SS**, then **→**, until `INVOICE2.VC` is on the edit line. Save it with the file name `INVOICE3.VC` by typing **ESCAPE ESCAPE ESCAPE ESCAPE 3.VC RETURN**.

## DISPLAYING THE FORMULAS OF THE WORKSHEET

Sometimes it's hard to remember just what formula you used at a particular location. If you decide to change a worksheet some time after you create it, it can take quite a while to scroll around and check each entry line, sometimes referring back and forth to compare related formulas. As usual, there's a simpler way.

The Expression option of the Attribute command displays what was entered in each cell, not the calculated value. After adjusting the width of some columns to display the full length of all formulas, you can quickly see the structure of the worksheet. Printing the worksheet in this form gives you a permanent copy of the structure, either for backup or as an aid to changing it.

Like all other attributes, the Expression option can be set globally. Type the following to see how wide the widest formula on your worksheet is:

```
>I1 RETURN
/GCC21 RETURN →
/GCC38 RETURN →
/GCC21 RETURN
```

Now type the following to set the Expression option globally:

```
/GAEY RETURN RETURN
```

Cells that contain labels or numbers don't change; cells that contain a formula, however, display the formula rather than the value of the formula. Figure 2-27 is a printed copy of the formulas of the worksheet; it represents the same worksheet whose values are shown in Figure 2-21.

		Midwest Hardware S 327 E. Mountain Av Berthoud, CO 80513		upply enuc		No: 201 Date: 10/16/83	
Bill to:				Ship to:			
Ranch Supply Co. 2025 S. Main Longmont, CO 80310				SAME			
Terms:	Net 30	Ship via:Truck	Resale #:	1234567	Cust. #:	200	
Item Number	Desc.	Quantity Ordered	Quantity Shipped	Retail Price	Disc. Price	Total Price	Back Ordered
20416	Gloves	100	10	@LOOKUP(E20,A4...A7)	+I20-(I20*(@LOOKUP(G20,C3...C10)/100)	+J20*H20	+G20-H20
20446	Glue	36	36	@LOOKUP(E21,A4...A7)	+I21-(I21*(@LOOKUP(G21,C3...C10)/100)	+J21*H21	+G21-H21
21329	Hammer	100	100	@LOOKUP(E22,A4...A7)	+I22-(I22*(@LOOKUP(G22,C3...C10)/100)	+J22*H22	+G22-H22
21377	Nails	50	50	@LOOKUP(E23,A4...A7)	+I23-(I23*(@LOOKUP(G23,C3...C10)/100)	+J23*H23	+G23-H23
Subtotal					@SUM(K20...K28)		
Shipping					@MAX(B16*K29,B17)		
Tax					@IF(J15>0,0,B15*K29)		
Total					@SUM(K29...K32)		
Amt. Rcv.							
Amt. Due					+K33-K34		

010-009

**Figure 2-27. Printing the Formulas of the Worksheet**

This gives you a clear picture of the structure of the worksheet. You can print this version of the worksheet the same way you print the calculated values; such a copy is valuable both as a reference for making changes and the safest possible backup.

## **SUMMARY**

By defining tabs and controlling the type of data that can be entered, you can set up data-entry applications to be used by others. More control over how entries are displayed and printed means you can tailor worksheets to meet specific needs, use them in presentations, or incorporate them into formal reports. Displaying or printing the formulas lets you quickly modify or correct a worksheet, or save a printed copy for fail-safe backup.

You have learned enough about the VisiCalc program to tackle a wide variety of applications, such as budgets, cash flow projections, investment portfolio evaluations, and investment strategy evaluations. You can focus your efforts on analysis instead of mechanical or clerical work, significantly increasing your productivity.

## LESSON SIX

Saving a Portion of the Worksheet	2-76
Defining Your Own Commands	2-78
Saving a DIF File Using a Keystroke Memory Sequence	2-80
Deleting a File	2-81
Resolving Forward References	2-81
Using a Worksheet as a Data-Entry Form	2-82
Using DIF Files to Create a Summary Report	2-84
Calculating an Average	2-85
Summary	2-87

---

Now that you have a VisiCalc worksheet to prepare invoices, it seems only natural to look for other ways the VisiCalc program can improve the information flow in a business. Summary sales reports, for example, provide a clear picture of the health of an enterprise.

This lesson shows you how to save a portion of a worksheet as a separate file that can be loaded into another worksheet. You'll use this feature to create a sales report from several invoices. The lesson also shows you how to create your own command sequences that can be repeated without reentering the sequence.

Figure 2-28 shows the entries to be added to the invoice.

Date	Invoice Number	Customer Number	Invoice Total	Shipping Charges	Total Items	Back Ordered
10/6/82	1024	191	205.70	15.00	76	90

**Figure 2-28. Invoice With Data Area To Be Saved**

Clear the worksheet and load the file you saved at the end of Lesson Five. If you are starting this lesson without having completed Lesson Five, load the file named Invoice3.VC from your VisiCalc program diskette.

## SAVING A PORTION OF A WORKSHEET

An option of the Storage command saves all or part of a VisiCalc worksheet in a form that makes it accessible to other worksheets or even other programs (such as the VisiSchedule™ program). This form is called DIF™, for Data Interchange Format.

When you save a data file, the VisiCalc program uses the cursor position as the upper-left corner of the area to be saved and asks you to specify the lower-right corner (just as when you print a file). Only the label or numeric value at each location is saved; formulas, formats, and attributes are not saved. This makes it possible for other programs to use the data, but it also means that a DIF file cannot be loaded as a regular worksheet.

To make it simple to save the data you want, you'll create a row of data that contains values from the invoice. For the sales report you'll create in a moment, you need the following items of data from the worksheet:

- Date
- Customer number
- Invoice subtotal (goods only, not shipping or tax)
- Shipping charges
- Total number of items
- Total back ordered

You can use the space below the lookup tables for this row of data. Enter the following:

```

/GAMD RETURN RETURN
> B33 RETURN
Invoice → Customer → Invoice → Shipping → Total → Back
RETURN
> A34 RETURN
Date → Number → Number → Total → Charges → Items →
Ordered RETURN
> A35 RETURN
/— — RETURN
/R RETURN → . → → → → → → RETURN ↓
/FL@Label(L5) → /FL+L3 → /FL+L15 → /F$+K33 →
/F$+K30 → /FL@SUM(H20.H28) → /FL@SUM(L20.L28) RETURN
    
```

Row 36 contains the data from the invoice you want to save for the summary sales report. Type the following to fill in the invoice locations referenced in the data area:

```

> L3 RETURN
1024 TAB
10/16/82 RETURN
> L15 RETURN
191 RETURN
> G22 RETURN
10 → 10 ← ↓ 20 → 20 RETURN
    
```

Type ! to recalculate the worksheet. The screen should look like Figure 2-29.

Item Number	Desc.	Quantity Ordered	Quantity Shipped	Retail Price	Disc. Price	Total Price	Back Ordered
09416	Gloves	100	10	5.90	3.54	75.40	30
09417	Glove	36	36	2.35	2.35	84.42	0
09418	Hammer	10	10	7.75	6.81	68.10	0
09419	Nails	20	20	.85	.84	12.70	0

010-020/P

**Figure 2-29. Data Area To Be Saved as a Separate File**

To save the data file, move the cursor to **A36** (the upper-left corner of the area to be saved), and type **/S#** (the **DIF** option of the Storage command). The prompt line reads `Data: Save Load;` type **S**.

Now the prompt line reads `Data save: File for Saving.` Data files are stored by the VisiCalc program in the Data Interchange Format (**DIF™**). You should always save data files with the extension `".DIF"` so you can identify them easily. Type **SLS1.DIF RETURN**. **SLS1** identifies the file as the first of several you'll save for the sales report, and **DIF** distinguishes it from a worksheet file.

The VisiCalc program must know the lower-right corner of the area to be saved.

The prompt line reads: `Data save: Lower right.` You can either move the cursor to the lower-right location, or type its coordinates; type **G36 RETURN**.

Finally, the prompt line reads `Data save: R, C or RETURN.` The VisiCalc program must know whether to save the data by rows (**R** or **RETURN**) or by columns (**C**). Usually you will want to save by rows, so type **R**. The *values* of **A36** through **G36** (*not* the formulas) are saved as file **SLS1.DIF**.

## DEFINING YOUR OWN COMMANDS

You're going to save four more sets of invoice data for the sales report. In each case, you must enter the same sequence of commands. As you may have guessed, the VisiCalc program offers a quicker way to do this repetitive task. You can define any series of keystrokes as a command whose name is a single character, then use that single character for the entire sequence of keystrokes.

Again, it's easier to do it than read about it. Type **/K**. The prompt line reads `KS MEMORY: = C E or RETURN.` **KS Memory (Keystroke Memory™)** is the VisiCalc program's term for a command you define by entering a series of keystrokes. The characters following the colon are options; they are described below.

- = Defines a sequence.
- C Clears, or erases, all sequence definitions.
- E Edits a sequence definition.
- RETURN Displays the sequence definitions.

Defining a sequence puts a series of keystrokes into the computer memory for recall at any time you need them. Type **=** to define a sequence. The prompt line reads *Sequence Name: A - Z*; the VisiCalc program wants you to name the sequence you're about to define. The name can be any letter; type **A**.

The prompt line now reads *Sequence Definition*, and the edit line reads *A:* followed by the edit cue. You now can type the keystrokes that define the sequence named **A**. Because typing an arrow key or RETURN ends the definition, you must type the character combinations shown below to represent the arrow keys and RETURN.

Character Combination	Represents
<b>^U</b>	Up arrow
<b>^D</b>	Down arrow
<b>^&lt;</b>	Left arrow
<b>^&gt;</b>	Right arrow
<b>^R</b>	RETURN

A few other combinations are used in sequence definitions; see "Keystroke Memory Command" in the Reference for a complete description and for an illustration on "Defining a Keystroke Memory Sequence."

In general, a sequence definition can include any number, formula, label, or command. Try it by defining a simple sequence that moves the cursor to **A25**, writes the label *Sales*, moves one location to the right, and writes the value *100*. Type the following:

```
> A25^R           [ Goto A25 ]
Sales^>          [ Write Sales followed by a right arrow ]
100^R RETURN    [ Write 100 ]
```

You have defined a sequence named **A** and it is now stored in memory ready to be used. To use a sequence, type **CONTROL-K**. The prompt line reads *KS Memory: Name # =*. Type **A** (the name of the sequence you just defined). Watch the prompt line, edit line, and worksheet; it's as if you typed the keystrokes in the definition.

Blank **A25** and **B25**; then delete the sequence by typing **/KCY**.

## SAVING A DIF FILE USING A KEYSTROKE MEMORY SEQUENCE

Sequence definitions are saved with the worksheet, so define a sequence named S (Save) that saves the data area of the worksheet. Type the following:

```
/K=S
!           [ Recalculate ]
> A36^R    [ Go to A36 ]
/S#S RETURN [ Save option of the Storage command ]
```

**NOTE:** A different form of the Keystroke Memory command records keystrokes as you type actual commands and write on the worksheet; the commands are carried out and the keystrokes are recorded at the same time. This technique doesn't use character combinations to represent the arrow keys or RETURN. See "The Define Sequence Option" under "Keystroke Memory Command" in the Reference for more details.

Testing the S sequence saves a data file with incorrect values, but you'll delete the file in a moment. Press **CONTROL-K S**. When the program asks for the file name, type **TEST.DIF RETURN**. The next prompt is Data Save: Lower right; type **G36 RETURN**. When the program asks whether to save by row (R) or column (C), type **R** or **RETURN**. After a few moments, the drive activity ceases and the prompt and edit lines clear.

If the computer beeps an error, strange entries appear on the worksheet, or other symptoms indicate that the sequence didn't work right, you can either edit or reenter the sequence definition. To edit it, type **/KES**; the definition appears on the edit line, and you can edit it just as you would edit an entry. To reenter it, type **/K=S**; when the prompt line asks you Sequence exists: Y to replace, type **Y**; then type the definition again.

The S sequence completes the invoice worksheet. Save it by typing **/SSINVOICE4.VC RETURN**.

## DELETING A FILE

To delete a file, type **/SD** (the Delete option of the Storage command). The prompt line reads *Storage: File to Delete*. Type **TEST.DIF RETURN** (the data file you saved a moment ago with the S sequence). The prompt line reads *Delete File: Type Y to confirm*. Type **Y**; the file is deleted from the disk.

Instead of typing the file name, you can also use **→** to display the file names. When the name of the file to be deleted is on the edit line, type **RETURN**.

## RESOLVING FORWARD REFERENCES

The worksheet is calculated from the upper-left corner downward and to the right; if a formula refers to a location below it or to its right, the value of the reference is calculated at the time the *referring* location is calculated; if the value of the referred-to location changes when it is calculated, the earlier reference won't be correct. Such a condition is called a *forward reference*.

The S sequence starts with **!** to recalculate the worksheet, because the locations in the data area (A36 through G36) refer to invoice locations to their right. The data area won't show the correct values unless you recalculate the worksheet after the last change to the invoice. Take a moment to see the effect of a forward reference.

Scroll the worksheet until row 36 is two or three rows above the bottom of the screen. Move the cursor to A32, and type **/WH**. A white bar between rows 31 and 32 divides the screen into two windows. The cursor is in the top window; type **>K33 RETURN**. Both the invoice total (in the lower-right corner of the top window) and the corresponding value in the bottom window (D36) read 205.70.

Move the cursor to H22 (Quantity Shipped of hammers), and type **50 RETURN**. The invoice total in the top window changes to 444.35, but in the data area (bottom window) it still reads 205.70. Now type **!**; the worksheet is recalculated, and the total in the data area changes to 444.35.

Forward references can be confusing. If a worksheet doesn't seem to be giving you correct results, check the formulas at the locations that are wrong; you'll often find a forward reference. You can usually correct the situation by moving some entries, changing the order of recalculation, or adding an intermediate value.

A forward reference can be dangerously misleading. But you can resolve the misleading calculation by manually recalculating the worksheet. Type **!** to recalculate the worksheet as many times as necessary until no entry changes value and the worksheet is stabilized.

With the cursor at H22, type **/B RETURN !/W1** to restore the worksheet.

If someone else is going to fill in the invoices, the data area could be distracting. Type the following to hide the data area:

```
> A33 RETURN
/AHY RETURN RETURN
/R(A RETURN RETURN → . G33 RETURN
/R(A RETURN G33 RETURN ↓ . ↓ ↓ ↓ RETURN
```

You can save this version to be used by someone else. (Remember, you earlier saved the worksheet with the tables visible, so you have a reference copy that shows all entries.) You may want to save this file under the name Invoice5.VC.

## USING A WORKSHEET AS A DATA-ENTRY FORM

Earlier you saved one data file (SLS1.DIF) from an invoice. Now enter several more invoices so you can save the summary data for each one. Enter only those locations saved in the data file. Type the following entries:

*Second Data File*

```
> L3 RETURN
1025 RETURN
> L15 RETURN
147 RETURN
> G20 RETURN
25 → 25 ← ↓ ↓
0 → 0 ← ↓
200 → 150 RETURN
CONTROL-K S
SLS2.DIF RETURN G36 RETURN
R
```

*Third Data File*

> L3 RETURN  
1026 RETURN  
> L15 RETURN  
212 RETURN  
> G20 RETURN  
0 → 0 ← ↓  
75 → 75 ← ↓  
20 → 20 ← ↓  
350 → 0 RETURN  
CONTROL-K S  
SLS3.DIF RETURN G36 RETURN  
R

*Fourth Data File*

> L3 RETURN  
1027 TAB  
10/17/82 RETURN  
> L15 RETURN  
147 RETURN  
> G20 RETURN  
10 → 10 ← ↓  
25 → 25 ← ↓  
10 → 10 ← ↓  
75 → 75 RETURN  
CONTROL-K S  
SLS4.DIF RETURN G36 RETURN  
R

*Fifth Data File*

> L3 RETURN  
1028 TAB  
10/18/82 RETURN  
> L15 RETURN  
223 RETURN  
> G20 RETURN  
0 → 0 ← ↓  
0 → 0 ← ↓  
0 → 0 ← ↓  
50 → 40 RETURN  
CONTROL-K S  
SLS5.DIF RETURN G36 RETURN  
R

You have saved five data files, SLS1.DIF through SLS5.DIF. Now you can create the sales report worksheet. You can load these files into this worksheet to create a summary report.

**USING DIF FILES TO CREATE A SUMMARY REPORT**

Figure 2-30 shows the sales report worksheet you will enter.

Midwest Hardware Supply Sales Summary Report -- October 1982						
Date	Invoice Number	Customer Number	Invoice Total	Shipping Charges	Total Items	Back Ordered
10/16/82	1024	191	205.70	15.00	76	90
10/16/82	1025	147	279.17	15.00	211	50
10/16/82	1026	212	294.56	15.00	95	350
10/17/82	1027	147	217.44	15.00	120	0
10/18/82	1028	223	37.10	15.00	40	10
Total			1033.97	75.00	542	500
Average			206.79	15.00	108.4	100

010-011

**Figure 2-30. Sales Report Worksheet for Data Files**

The report consists of the title, column headings, format attributes for 10 rows into which you can load the data files, and a row of totals and averages at the bottom. Clear the worksheet, and type the following to put the title and column headings on the summary report:

```
>E1 RETURN
Midwest H → ardware S → upply ← ← ← ↓
Sales Sum → mary Repo → rt -- Oct → ober 1982 RETURN
>B4 RETURN
Invoice → Customer → Invoice → Shipping → Total → Back
RETURN
>A5 RETURN
Date → Number → Number → Total → Charges → Items → Ordered
RETURN
>A6 RETURN
/-- RETURN
/R RETURN → .G6 RETURN
```

The column headings match the items you saved from each invoice. You will load one data file per row beneath the column headings. The first row under the data rows totals the Invoice Totals, Shipping Charges, Total Items, and Back Order. The second row averages them.

## CALCULATING AN AVERAGE

The Average function can be used to find the average value for each column. But first, the rows for the data files need to be formatted. Move the cursor to A7 and type the following:

```
/GAV < RETURN RETURN → → → /F$ → /F$ ←  
/R(A RETURN → RETURN ↓ . D16 RETURN
```

Now enter the totals using the Sum function and enter the averages using the Average function. The Average function divides the *sum* of its arguments by the *number* of non-zero arguments. Type the following:

```
> A17 RETURN  
/— = RETURN  
/R RETURN → .G17 RETURN ↓  
Total → → → /F$@SUM(D6.D17) → /F$@SUM(E6.E17) →  
@SUM(F6.F17) → @SUM(G6.G17) RETURN  
> A19 RETURN  
/— — RETURN  
/R RETURN → .G19 RETURN ↓  
AVERAGE → → → /F$@AVERAGE(D6.D17) →  
/F$@AVERAGE(E6.E17) → @AVERAGE(F6.F17) →  
@AVERAGE(G6.G17) RETURN
```

The summary report is ready to have the invoice data loaded into it. Remember, when you *save* a data file, you must position the cursor at the upper-left corner of the rectangle to be saved, then specify both the lower-right corner and whether to save the data by rows or by columns. Similarly, to *load* a data file, you must position the cursor at the upper-left corner of the area into which it is to be loaded and specify whether to load by rows or columns. As when saving a data file, you load data by rows except in unusual circumstances.

Before you begin loading data files, save the worksheet. This way, you will have a skeleton worksheet (called a template) that you can use many times as the basis for sales reports. Type **/SSSalesrepo.VC RETURN**.

Move the cursor to A7 to load the data file into row 7, and type the following:

```
/S#L
SLS1.DIF RETURN
R
```

The numbers appear in the first row of data. The screen should look like Figure 2-31.

Date	Invoice Number	Customer Number	Invoice Total	Shipping Charges	Total Items	Back Ordered
10/15/82	1024	191	205.70	15.00	76	90
Total			205.70	15.00	76	90
Average			205.70	15.00	76	90

010-021/P

**Figure 2-31. Summary Sales Report With One Data Row**

To load the remaining data files, type the following:

```
↓ /S#L
SLS2.DIF RETURN
R ↓
/S#L
SLS3.DIF RETURN
R ↓
/S#L
SLS4.DIF RETURN
R ↓
/S#L
SLS5.DIF RETURN
R
```

The sales report now includes all the data files you saved from the invoices, and shows the totals and averages. It should look like Figure 2-30.

## **SUMMARY**

This lesson showed you how to move data back and forth between worksheets, and how to define your own commands to speed building a worksheet or make it easier for someone else to use your worksheet.

Now you can design a set of worksheets that use common data. Different departments can use the same basic worksheet, for example, to forecast next year's budget; a summary budget worksheet can be built from the totals row saved from each department budget. A purchasing forecast worksheet can be built from summary data saved from worksheets used by Sales, Service, Manufacturing, and Finance. Using Keystroke Memory sequences, you can also accumulate numbers in one part of a worksheet as you overlay different sets of data in another part of the same worksheet, and thus create the kind of summary reports you need for your business.

## LESSON SEVEN

Using Keystroke Sequences for Common Worksheet Features	2-88
Editing a Sequence Definition	2-89
Defining a Sequence Usable Anywhere on the Worksheet	2-89
Building a Worksheet Fast With Keystroke Memory Sequences	2-91
Replicating a Rectangular Area	2-91
Moving Rows and Columns	2-92
An Income Variance Report	2-93
Setting Manual Recalculation	2-95
Summary	2-97

---

The last several lessons dealt with an invoice. This lesson shows how you can use Keystroke Memory commands to enter a worksheet quickly, once you have planned it, and introduces a new example — an income variance report — to show more features of the Replicate command.

### USING KEYSTROKE SEQUENCES FOR COMMON WORKSHEET FEATURES

If you use a particular worksheet frequently, you can save yourself considerable time by using Keystroke Memory sequences to help you build the worksheet. First, define the sequence commands most frequently used. Then, save them on a blank worksheet. Finally, when you build the worksheet, load the worksheet that contains the sequence definitions, and use the sequences to build the worksheet.

Consider a budget, for example. Usually, you write column headings for the name of the month, total, and a line of dashes to separate the column headings from the budget entries.





## **Building a Worksheet Fast With Keystroke Memory Sequences**

Another feature of keystroke sequences makes it possible to use them to build a worksheet fast: you can repeat a sequence by typing the number of times it should be carried out before typing the name of the sequence. For example, if you wanted sequence I to be repeated four times, you would type CONTROL-K 4I.

To demonstrate how much the sequences can speed your work, enter the branch office budget you entered in Lesson Two. Start by typing /**CAY** to clear the worksheet.

Type the following to enter the first block of the budget:

```
/SLKSBUDGET.VC RETURN  
>A5 RETURN  
/GCC14 RETURN  
CONTROL-K H ↓  
Secretary ↓ Accts. Clerk ↓ Receptionist ↓ Office Mgr. ↓  
CONTROL-K L →  
@SUM(↑↑↑↑↑.↑) RETURN  
CONTROL-K M  
>A6 RETURN  
CONTROL-K 4I  
>A6 RETURN  
CONTROL-K 4Y  
>B7 RETURN  
1000 ↓ 850 ↓ 800 ↓ 1250 RETURN
```

There's the first block of your budget; what about the second? Its structure is the same as the first block. The labels and numbers are different, but those are easy to change. Not surprisingly, the Replicate command gives you a way to capitalize on this similarity and make your work easier.

## **REPLICATING A RECTANGULAR AREA**

Just as you can replicate a single cell, a row, or a column, you can replicate a block of the worksheet by specifying the upper-left and lower-right corners as the source range. In this example, all references to other locations are to be replicated Relative to their new location. You can specify this by typing a left parenthesis after the /R to identify a replicate option, then R for Relative. (You can also, of course, replicate all references with No change by typing N after the left parenthesis.)

**WARNING:** Be sure that the area to receive the replication contains no other entries! The Replicate command does *not* insert rows or columns to accommodate the new entries. The entries being replicated overwrite any entries in the area.

To create the second block of the budget, type the following:

```
> A6 RETURN
/R(R RETURN .N12 RETURN A15 RETURN
> A14 RETURN
Phys. Plant ↓ ↓ Rent ↓ Utilities ↓
Insurance ↓ Maintenance RETURN
> B16 RETURN 1300 ↓ 200 ↓ 400 ↓ 150 RETURN
```

And there's the second half of your budget. Compare this with Figure 2-6.

You could quickly add additional categories to this budget, then add the grand totals at the bottom of the sheet. With these few sequences, preparing an annual budget is easy. This is what computers are for, to do the mechanical work and let you concentrate on analysis.

## MOVING ROWS AND COLUMNS

If you want to rearrange a worksheet, the Move command lets you move entire rows or columns. As when you insert or delete a row or column, all references on the worksheet to cells in the moved entries are corrected.

Suppose you want to move the Maintenance row above the Utilities row, so that it follows the Rent row. Move the cursor to A19 and type /M. The prompt line reads Move: From: R C or Range. Type R.

Now the edit line reads R:, followed by the edit cue; the program is ready for you to specify where to move the row. The Move command moves a row *above* the destination, so move the cursor up to Utilities (you could also type A17) and press RETURN. The row is moved.

You can also move a range of rows. To move Receptionist and Office Mgr. to the top of the first block, type the following:

```
> A9 RETURN
/M2R ↑ ↑ RETURN
```

Moving columns works the same way. The moved columns are placed to the *left* of the destination you specify.

You saved this budget in Lesson Two, and you saved the sequences earlier in this lesson. Clear the worksheet (but keep the sequence definitions) by typing **/CSY**.

## AN INCOME VARIANCE REPORT

After you forecast income, you often need to track actuals against the forecast. The VisiCalc program makes it easy to record the forecast and actual income, then calculate the variance between them.

Figure 2-32 shows the income variance report you will enter.

```
Income Variance Report      1982
Date of forecast 1/25/83
Month of report: 7
```

Forecast	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Total
Williams Realty	2,500	2,500	2,500	2,500	2,500	2,500	2,500	2,500	2,500	2,500	2,500	2,500	30,000
State of Del.	1,200			1,200			1,200			1,200			4,800
Fortech Eng.			1,500	2,000	3,000								6,500
Edgewater Lodge			1,000										1,000
Bell State Coll.										3,000			3,000
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----
Monthly Total	3,700	2,500	5,000	5,700	5,500	2,500	3,700	2,500	2,500	6,700	2,500	2,500	45,300
Cumulative Total	3,700	6,200	11,200	16,900	22,400	24,900	28,600	31,100	33,600	40,300	42,800	45,300	45,300
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----
Checks Received													
Williams Realty	2,000	2,200	2,500	2,500	2,500	2,500	2,500						16,700
State of Del.	600	600	600	1,200			1,200						4,200
Fortech Eng.			2,000	2,000	2,500	2,500							9,000
Edgewater Lodge			450										450
Bell State Coll.													0
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----
Monthly Total	2,600	2,800	5,550	5,700	5,000	5,000	3,700	0	0	0	0	0	30,350
Cumulative Total	2,600	5,400	10,950	16,650	21,650	26,650	30,350	30,350	30,350	30,350	30,350	30,350	30,350
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----
Variance													
Williams Realty	(500)	(300)	0	0	0	0	0	0	0	0	0	0	(800)
State of Del.	(600)	600	600	0	0	0	0	0	0	0	0	0	600
Fortech Eng.	0	0	500	0	(500)	2,500	0	0	0	0	0	0	2,500
Edgewater Lodge	0	0	(550)	0	0	0	0	0	0	0	0	0	(550)
Bell State Coll.	0	0	0	0	0	0	0	0	0	0	0	0	0
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----
Monthly Total	(1,100)	300	550	0	(500)	2,500	0	0	0	0	0	0	1,750
Cumulative Total	(1,100)	(800)	(250)	(250)	(750)	1,750	1,750	1,750	1,750	1,750	1,750	1,750	1,750

010-012

**Figure 2-32. Income Variance Report**

To enter the titles, column headings, and row labels, type the following:

```

/GCC16 RETURN
Income Variance → Report → " 1982 ↓ ← ←
Date of forecast → " 1/25/83 ↓ ←
Month of report: → /FL7 ↓
/AHY RETURN RETURN 1 → /AHY RETURN RETURN + ← + 1
RETURN
/R RETURN → .M4 RETURN R ↓ ← ←
Forecast CONTROL-K H ↓
Williams Realty ↓ State of Del. ↓ Fortech Eng. ↓
Edgewater Lodge ↓ Bell State Coll. ↓
CONTROL-K L ↓
Monthly Total → @Sum( ↑ ↑ ↑ ↑ ↑ ↑ ↑
. ↑ ) RETURN
CONTROL-K M ↓ ←
Cumulative Total → + ← + ↑ RETURN
CONTROL-K M
> A6 RETURN
CONTROL-K 5Y
> N14 RETURN
/B RETURN

```

Blanking the entry at N14 is needed because the Keystroke sequence creates a total entry for the year that you don't need. It's quicker to use the Keystroke sequence to replicate the monthly totals and erase this extraneous entry than it is to replicate the monthly totals without the yearly total.

The entries in this block don't refer to the previous month's entry, because you will only enter income in the month you expect to receive it. There should be no entry in those months in which you expect no income from a given client.

The next two blocks of the report have the same structure, but are used to enter checks received and calculate the variance. Type the following:

```

> A6 RETURN
/R(R RETURN .N15 RETURN A18 RETURN
> A17 RETURN
Checks Received RETURN
/R(R RETURN .N28 RETURN A31 RETURN
> A31 RETURN
Variance RETURN

```

The worksheet should look like Figure 2-32.

## SETTING MANUAL RECALCULATION

Unless you specify otherwise, the worksheet is recalculated automatically each time you write a value on the worksheet. Sometimes this can delay you, particularly if the worksheet contains many functions or lengthy formulas. The VisiCalc program lets you specify whether recalculation is automatic or manual.

Type **/GR** (the Recalculate option of the Global command). The prompt line reads `Global: Recalc: A M` (the options are **A**utomatic and **M**anual). Type **M** to specify manual recalculation. Now the worksheet is recalculated only when you type an exclamation point (!). You can change back to automatic recalculation any time by typing **/GRA**.

The worksheet needs the forecasted income and checks received in the top two blocks. Move the cursor to **B7** and enter the numbers shown in Figure 2-32. (Don't enter the commas.)

The third block needs a formula in each location to calculate the variance. In its simplest form, it subtracts actual income from forecasted income, month-by-month. Because each location contains the same formula, you need enter it only once; you replicate it down the column of clients, then replicate this column across the rows of months. Type the following:

```
> B33 RETURN
+ B19 - B7 RETURN
/R RETURN ↓ . ↓ ↓ ↓ ↓ RETURN RR
/R(R RETURN ↓ ↓ ↓ ↓ RETURN → . M33 RETURN
/GAVI,( RETURN RETURN
```

The last entry specifies that all values are to be displayed as integers, with a comma marking thousands, and negative values enclosed in parentheses. Type **!** to recalculate the worksheet, then scroll across and look at the numbers. Figure 2-33 shows the worksheet as you have entered it.

Income Variance Report 1982  
 Date of forecast 1/25/83  
 Month of report: 7

Forecast	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Total
Williams Realty	2,500	2,500	2,500	2,500	2,500	2,500	2,500	2,500	2,500	2,500	2,500	2,500	30,000
State of Del.	1,200			1,200			1,200			1,200			4,800
Portech Eng.			1,500	2,000	3,000								6,500
Edgewater Lodge			1,000										1,000
Bell State Coll.										3,000			3,000
Monthly Total	3,700	2,500	5,000	5,700	5,500	2,500	3,700	2,500	2,500	6,700	2,500	2,500	45,300
Cumulative Total	3,700	6,200	11,200	16,900	22,400	24,900	28,600	31,100	33,600	40,300	42,800	45,300	45,300
Checks Received													
Williams Realty	2,000	2,200	2,500	2,500	2,500	2,500	2,500						16,700
State of Del.	600	600	600	1,200			1,200						4,200
Portech Eng.			2,000	2,000	2,500	2,500							9,000
Edgewater Lodge			450										450
Bell State Coll.													0
Monthly Total	2,600	2,800	5,550	5,700	5,000	5,000	3,700	0	0	0	0	0	30,350
Cumulative Total	2,600	5,400	10,950	16,650	21,650	26,650	30,350	30,350	30,350	30,350	30,350	30,350	30,350
Variance													
Williams Realty	(500)	(300)	0	0	0	0	0	(2,500)	(2,500)	(2,500)	(2,500)	(2,500)	(13,300)
State of Del.	(600)	600	600	0	0	0	0	0	0	(1,200)	0	0	(600)
Portech Eng.	0	0	500	0	(500)	2,500	0	0	0	0	0	0	2,500
Edgewater Lodge	0	0	(550)	0	0	0	0	0	0	0	0	0	(550)
Bell State Coll.	0	0	0	0	0	0	0	0	0	(3,000)	0	0	(3,000)
Monthly Total	(1,100)	300	550	0	(500)	2,500	0	(2,500)	(2,500)	(6,700)	(2,500)	(2,500)	(14,950)
Cumulative Total	(1,100)	(800)	(250)	(250)	(750)	1,750	1,750	(750)	(3,250)	(9,950)	(12,450)	(14,950)	(14,950)

010-013

**Figure 2-33. Inaccurate Variance Report**

Things look pretty good until you get to July; then something goes wrong. The date of the report is July (month 7, entered at B3), so no income is entered in the following months. But the formula in the variance block continues to subtract forecasted income from actual income, producing negative variances in all entries where you forecasted income from August through December. This gives an inaccurate and bleak picture for the year, prevents the calculation of accurate year-to-date totals, and makes it difficult to pick out the year-to-date cumulative total (it's the Cumulative Total entry for July).

You need a formula that checks the current month against the month of the report, and calculates a variance only if the current month is equal to or less than the month of the report. The current month number is in row 4 (hidden for aesthetic reasons), and the month of the report is at B3.

Move the cursor to B33, and type the following:

**@If(B4 <= B3, + B19 - B7, 0) RETURN**

You need to replicate this formula down the column, then replicate the column across the rows. Specifying whether each reference to another location is Relative or No change must be done individually for each reference.

When you replicate down the column, the references to the month of the report and the current month are unchanged (N); the references to actual and forecasted income are relative (R). Type the following:

**/R RETURN ↓ . ↓ ↓ ↓ ↓ RETURN NNRR**

The reference to the current month must be Relative, however, as you replicate across the columns of months. The sequence is RNRR for each of the five rows you're replicating; type the following:

**/R. ↓ ↓ ↓ ↓ RETURN → . M33 RETURN  
RNRR  
RNRR  
RNRR  
RNRR  
RNRR  
!**

Scroll across and check the numbers. All entries after July are 0. The Total column (after December) contains accurate year-to-date numbers. The variance report is complete and accurate.

## **SUMMARY**

Keystroke sequences, the Replicate command, and some planning can dramatically reduce the time it takes you to build a worksheet. Not only is the VisiCalc program powerful, it's fast. That's what personal computing means: the power of a computer, under your control, focused on your needs, subject to your priorities. The VisiCalc program gives you that control.

## LESSON EIGHT

Finding the Future Value of a Series of Payments	2-100
Finding the Present Value of a Series of Payments	2-102
Finding the Payment of a Loan	2-103
Finding the Future Value of a Fixed Amount	2-104
Finding the Interest Between Two Dates	2-105
Finding the Present Value of Unequal Payments	2-106
Finding the Internal Rate of Return	2-107
Summary	2-108

---

Financial time-lines are common applications of the VisiCalc program — the original purpose, in fact, for which it was designed. Formulas, separate tables for looking up interest rates and expense ratios, the Replicate command, keystroke sequences, and other features of the program help you to quickly construct worksheets that handle budgeting, annuity, amortization, and other financial systems.

But many financial calculations involve equal payments (or cash flows, or annuities, or other payments or receipts), and don't require an entry for each time period. For some problems, in fact, it is neither necessary nor practical to build a time line. Frequently all you're interested in is the future value or number of periods required.

So the VisiCalc program includes a set of financial functions that solve equal-payment rate-term-value problems. Each function requires a value for only three of the four possible variables: interest rate, payment, term (number of periods), present value, and future value. When a financial function takes term ( $n$ ) and/or interest rate ( $i$ ) as arguments, they must be specified.

The financial functions calculate values to a greater precision than many calculators, however, so be aware that you may need to round your results to make them agree with those obtained with a calculator (see the section on "The Financial Functions" in the VisiCalc Reference, Chapter 3).

The following table lists the financial functions and the arguments they require.

<b>Function</b>	<b>Name and Arguments</b>
Present Value	@PV(rate,term,payment,future value).
Future Value	@FV(rate,term,payment,present value).
Rate	@RATE(term,payment,present value,future value).
Term	@PERIODS(rate,payment,present value,future value).
Payment	@PMT(rate,term,present value,future value).

**Notes:**

rate	Interest rate per period.
term	Number of periods.
payment	Amount of the payment.
present value	Present value of the transaction.
future value	Future value of the transaction.

The VisiCalc program also includes two functions that solve financial problems for unequal payments: Net Present Value (@NPV) and Internal Rate of Return (@IRR). These are described later in this lesson.

In this lesson you will enter a worksheet that solves financial problems: payment required to amortize a loan, future value of a series of payments, present value of an annuity, etc. Figure 2-34 shows the worksheet you will enter.

Deposit	500												
Interest Rate	.13												
Term	24												
-----													
Future Value	13,621												
Payment	500												
Interest Rate	.13												
Term	24												
-----													
Present Value	10,517												
Sum of payments	12,000												
Amount	5,000												
Interest Rate	.18												
Term	36												
-----													
Payment	180.76												
Initial Balance	5000												
Interest Rate	.135												
Term	60												
-----													
Value	9,783												
Revenue	5000	5,100	5,202	5,306	5,412	5,520	5,631	5,743	5,858	5,975	6,095	6,217	
Interest Rate	.13												
-----													
Total Revenue	67,060												
Present Value	62,409												
		Deposit	Withdr.										
		-----											
Month	9	12											
Day	1	31											
Year	1983	1983											
Amount	3000												
Interest Rate	.13												
-----													
Value	3132.09												
Initial Cost	50000												
Savings	10000	15000	15000	15000	15000								
Int. Rate Return	11.7%												

010-014

**Figure 2-34. Financial Calculator**

Appendix D describes the mathematics used to calculate the financial functions.

**NOTE:** The financial functions for unequal payments are based on the assumption that payments are always negative. The effect of this assumption will become clear as you enter the functions on the worksheet.

## **FINDING THE FUTURE VALUE OF A SERIES OF PAYMENTS**

The Future Value function tells you how much a series of deposits (or some other sort of payment) will be worth at some point in the future. It requires values for three of the following four arguments, in the order shown; the argument not provided is entered as @NA:

1. Interest rate (for the period; if the period is a month, for example, the interest rate is 1/12 the annual rate; this argument must be specified).

2. Number of periods (this argument must be specified).
3. Payment (always a negative value; this argument need not be specified if Present value is given).
4. Present value of the payments (this argument need not be specified if Payment is given).

Type the following:

```

/CAY
/GRM
/GCC17 RETURN
Deposit → 500 ← ↓
Interest Rate → .13 ← ↓
Term → 24 ← ↓
/— — → /— — ← ↓
Future Value → /AVI, RETURN RETURN
@FV( ↑↑↑ /12,
↑↑,
— ↑↑↑↑,
@NA) RETURN
    
```

Because the interest rate is expressed in annual terms (13 percent), the first argument (rate) is divided by 12. You could enter the annual interest rate as .01083333333333 (the monthly equivalent of 13 percent annual interest) and not divide the rate argument by 12, but it's easier and usually more natural to enter annual interest rates on the worksheet. All the examples in this chapter show annual rates on the worksheet; the financial functions convert to monthly equivalents, where required.

Because payments are always negative, the value of the third argument (payment) is preceded by a minus sign; this lets you enter the payment as a positive number, making the worksheet read more naturally.

As B5 shows, if you deposit \$500 per month at 13 percent interest, after two years you will have \$13,621. You might want to try some other values before going on; if you use some large numbers, you may have to make column B wider to display the future value. (The first thing you typed was /GRM, which sets manual recalculation, so remember to type ! to recalculate the worksheet after changing the values.)

## FINDING THE PRESENT VALUE OF A SERIES OF PAYMENTS

The Present Value function reverses the Future Value function: it tells you how much money in hand today is equivalent to a future series of payments. It requires values for three of the following four arguments, in the order shown; the argument not provided is entered as @NA:

1. Interest rate (for the period; if the period is a month, for example, the interest rate is 1/12 the annual rate; this argument must be specified).
2. Number of periods (this argument must be specified).
3. Payment (always a negative value; this argument need not be specified if Future value is given).
4. Future value of the payments (this argument need not be specified if Payment is given).

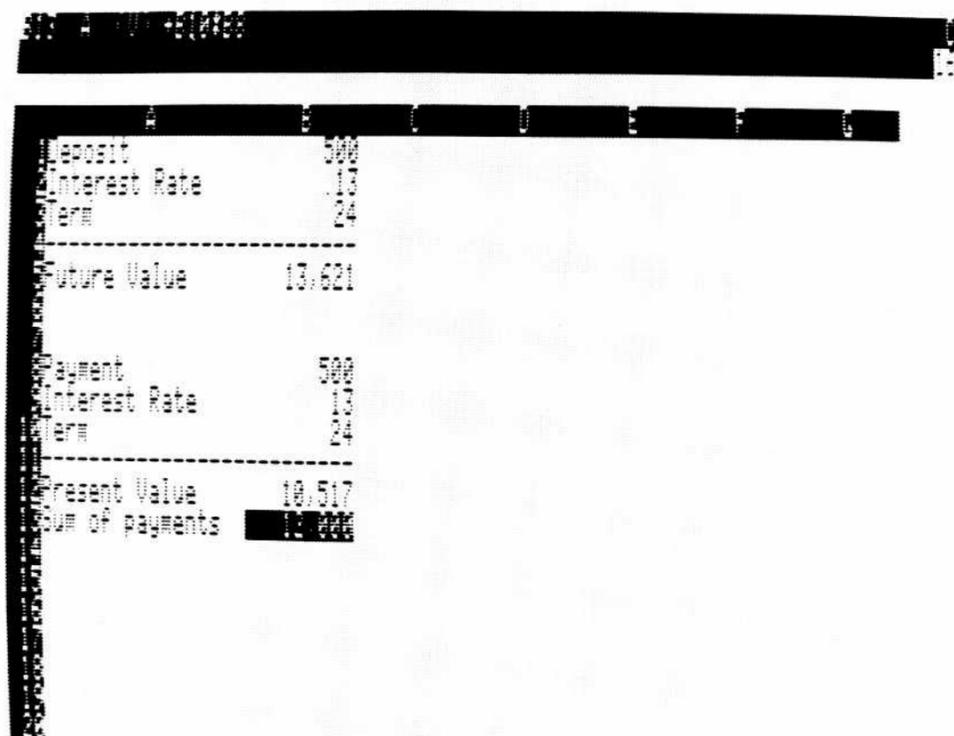
Type the following:

```
> A8 RETURN
Payment → 500 ← ↓
Interest Rate → .13 ← ↓
Term → 24 ← ↓
/— — → /— — ← ↓
Present Value → /AVI, RETURN RETURN
@PV( ↑ ↑ ↑ /12,
↑ ↑ ,
— ↑ ↑ ↑ ↑ ,
@NA) RETURN
```

Again, the interest rate is divided by 12 and the payment is preceded by a minus sign. Now B12 shows you that 24 monthly payments of \$500 are worth \$10,517 today. To see how these two functions give you a clearer picture of the time value of money, type the following:

```
> A13 RETURN
Sum of payments →
/AVI, RETURN RETURN
+ ↑ ↑ ↑ * ↑ ↑ ↑ ↑ ↑ RETURN
```

Make sure the payment, interest rate, and term of the future value calculation are 500, .13, and 24, respectively. The screen should look like Figure 2-35.



010-022/P

**Figure 2-35. Future Value and Present Value Calculations**

The sum of the payments is \$12,000. If you deposit those payments at 13 percent interest over two years, you'll have \$13,621. If someone pays you \$500 a month for two years, that's worth the same as paying you \$10,517 today if you could invest the money at 13 percent.

## FINDING THE PAYMENT OF A LOAN

The Payment function tells you what payment is required to amortize a debt. It requires values for three of the following four arguments, in the order shown; the argument not provided is entered as @NA:

1. Interest rate (for the period; if the period is a month, for example, the interest rate is 1/12 the annual rate; this argument must be specified).
2. Number of periods (must be specified).
3. Present value of the payments (not necessary to specify if Future value is given).
4. Future value of the payments (not necessary to specify if Present value is given).

Type the following:

```

> A16 RETURN
Amount → /AVI, RETURN RETURN
5000 ← ↓
Interest Rate → .18 ← ↓
Term → 36 ← ↓
/— — → /— — ← ↓
Payment → /F$
@PMT( ↑↑↑ /12,
↑↑,
— ↑↑↑↑↑,
@NA) RETURN

```

Because the function solves for payment, the function itself is preceded by a minus sign in this example so that the figure on the worksheet will be positive. If you want the figure to be negative, simply enter the function with no preceding minus sign.

If you borrow \$5000 at 18 percent interest and pay it back over three years, the monthly payment is \$180.76

The functions for finding interest rate (@RATE) and number of periods (@PERIODS) operate similarly; the table at the beginning of the lesson shows the arguments they require.

## FINDING THE FUTURE VALUE OF A FIXED AMOUNT

You can find the value of a fixed amount that draws interest by specifying the term, present value, and interest rate, and setting the payment to @NA. To find out what \$5000 will be worth at 13.5 percent interest in 5 years, type the following:

```

> A23 RETURN
Initial Balance → 5000 ← ↓
Interest Rate → 13.5% ← ↓
Term → 60 ← ↓
/— — → /— — ← ↓
Value → /AVI, RETURN RETURN
@FV( ↑↑↑ /12,
↑↑,
@NA,
— ↑↑↑↑↑ ) RETURN

```

In this case, the present value is negative because it is considered to be a single payment made at the beginning of the term. To make the future value positive, the Present Value argument is preceded by a minus sign. The future value of this example is 9,783.

## FINDING THE INTEREST BETWEEN TWO DATES

The VisiCalc program includes several functions that calculate dates and time. January 1, 1979 is used as the base date by the VisiCalc program for all these calculations. One of these functions — MDY, for month-day-year — is described here. For a complete description of the calendar and time functions, see "Functions" in the Reference.

The MDY function requires three arguments: month, day, and year. The value of the function is the number of days between the date specified by the arguments and the base date (January 1, 1979). The following examples show the value of several MDY functions:

@MDY(1,2,1979)	1
@MDY(1,1,1980)	365
@MDY(12,1,1978)	-31
@MDY(1,1,1979)	0

As the last example shows, the value of the MDY function that specifies the base date itself is zero. Be sure to enter the full year, not just the last two digits. The value of @MDY(1,1,79) is -693960, not 0, because 79 means the year 79, not 1979.

To calculate the interest on an amount based on the dates of deposit and withdrawal (rather than the number of periods), type the following:

> **A37 RETURN**

**Deposit Date** → @MDY(9,1,1983) ← ↓

**Withdrawal Date** → @MDY(12,31,1983) ← ↓

**Amount** → 3000 ← ↓

**Interest Rate** → .13 ← ↓

/— — → /— — ← ↓

**Value** → @FV(B40/365,B38-B37,@NA,-B39) RETURN

The term argument in the Future Value function is calculated by subtracting the earlier date (B37) from the later date (B38). The rate argument in the Future Value function is divided by 365 because the unit of time of the MDY function is days. The Future Value function shows you that if you deposit \$3,000 at 13 percent interest on September 1, 1983, it will be worth \$3,132.09 on December 31, 1983.

Entering dates in this format can be a bit awkward; it's fairly simple to modify the worksheet so you can enter the date in a more convenient form and let the Future Value function convert the entries to the proper form. Type the following:

```
> A37 RETURN
/B → /FRDeposit → /FRWithdr. ← ← ↓
/— → /— → /— ← ← ↓
/I4R
Month → 9 → 12 ← ← ↓
Day → 1 → 31 ← ← ↓
Year → 1983 → 1983 RETURN
> B46 RETURN
/E
```

This worksheet lets you enter the month, day, and year separately. The Future Value function is on the edit line; you must change it to calculate the date (the month, day, and year entries). Move the edit cue until it highlights the comma that follows B38–B37, type **ESCAPE** seven times to erase the term argument, then type the following:

```
@MDY(C39,C40,C41)-@MDY(B39,B40,B41) RETURN
```

The MDY functions calculate the date from the separate entries; the earlier date is subtracted from the later one, just as it was before. The future value (at B46) is still 3132.09.

## FINDING THE PRESENT VALUE OF UNEQUAL PAYMENTS

The functions that deal with a series of unequal payments use a different technique for specifying the data. Rather than specifying each item of data as an argument, you provide only two arguments: one key item of data, and the range of locations that contains a series of payments (or receipts).

The Net Present Value (@NPV) function, for example, requires the following two arguments:

1. Discount rate (or interest, or cost of money, whichever term you prefer), expressed in terms appropriate to the length of the period.
2. The starting and ending locations of the range (either row or column) that contains the amounts.

Assume, for example, you forecast net revenue by month for the coming year. What is the net present value of that revenue? If each month's revenue is the same, you could use the Present Value (@PV) function. But what if the monthly revenues vary? Type the following:

**> A30 RETURN**

**Revenue** →

**/F15000** → **/AVI, RETURN RETURN + ← \* 1.02 RETURN**

**/R RETURN** → **. M30 RETURN**

**R** ← ← ↓

**Interest Rate** → **.13** ← ↓

**/— —** → **/— —** ← ↓

**Total Revenue** → **/AVI, RETURN RETURN @SUM(B30.M30)** ← ↓

**Present Value** → **/AVI, RETURN RETURN**

**@NPV( ↑ ↑ ↑ /12, B30.M30) RETURN**

As usual, the interest rate is entered on the worksheet in annual terms, then divided by 12 to produce the monthly equivalent. The Net Present Value function doesn't assume that payments are negative, so no correction is made. The formula that you replicated across 12 months increases revenue by 2 percent a month. This makes the twelfth month's revenue 6,217 (at M30), compared to the original month's 5,000 (at B30).

The Net Present Value function discounts the first period once by the specified rate, the second period twice, the third period three times, etc. The total revenue for the year is 67,060, equivalent to receiving \$62,409 at the beginning of the year (the net present value).

## **FINDING THE INTERNAL RATE OF RETURN**

The Internal Rate of Return (IRR) function calculates the effective return on a capital purchase, development project, or other internal investment. It is a special case of net present value; the internal rate of return is the interest rate that discounts a series of unequal payments to a present value equal to the negative of the initial investment. This allows you to compare the return on the investment to the return available by investing the same amount of money at the prevailing interest rate.

Suppose the Manufacturing division of your company proposes to install a \$50,000 packaging machine. They estimate it will save \$10,000 the first year, and \$15,000 a year for the next four years. What is the internal rate of return of the proposed investment? Type the following:

> A49 RETURN

Initial Cost → 50000 ← ↓

Savings → 10000 → 15000 RETURN

/R RETURN → . → → → RETURN

> A51 RETURN

Int. Rate Return → /AV1F% RETURN RETURN

@IRR(-B49,B50.F50) RETURN

The initial cost is preceded by a minus sign in the IRR function because it is considered a single payment. The internal rate of return of this investment is 11.7%. You can compare this figure to prevailing interest rates to help evaluate the proposed investment.

Save this worksheet by typing /SSFINANCIAL.VC RETURN.

## SUMMARY

The financial functions save hours of calculations or looking up values in a book of tables. The VisiCalc program makes it easy for you to incorporate the time value of money into planning worksheets. With the increasing importance of interest rates, you needn't rely on others to provide this information or make a possibly inaccurate approximation; you have direct control over the accuracy, availability, and use of that information.

## LESSON NINE

A Cost-Estimating Worksheet	2-111
Modulo Arithmetic	2-114
Integer Arithmetic	2-114
Drawing Bar Graphs	2-117
Summary	2-118

---

In addition to the financial functions, the If and Lookup functions, and the Sum function, the VisiCalc program includes more arithmetic functions that can be used in formulas to simplify your calculations. You can also use an option of the Format command to draw simple bar graphs on a worksheet.

The table below lists the arithmetic functions and the arguments they require.

This lesson uses a few examples of the arithmetic functions in a simple cost-estimating worksheet for material fabrication. For a more detailed description of the purpose and requirements of each function, see "Functions" in the Reference.

Function	Name and Arguments
Absolute value	@ABS(value)
Average	@AVERAGE(list)
Count	@COUNT(list)
Dot product	@DOTPROD(range1,range2)
Exponential value	@EXP(value)
Integer part	@INT(value)
Natural logarithm	@LN(value)
Logarithm	@LOG10(value)
Minimum	@MIN(list)
Maximum	@MAX(list)
Modulo arithmetic	@MOD(value,modulus)
Pi	@PI
Square root	@SQRT(value)
Sum	@SUM(list)
Sine	@SIN(value)
Cosine	@COS(value)
Tangent	@TAN(value)
Arcsine	@ASIN(value)
Arccosine	@ACOS(value)
Arctangent	@ATAN(value)

**NOTES:**

1. *value* is any VisiCalc value.
2. *list* is a series of values separated by commas.
3. *range* is the starting and ending location of a range of locations, separated by an ellipsis (...).
4. The values for the trigonometric functions (Sine, Cosine, etc.) must be specified in radians. To convert degrees to radians, multiply by .0174533; to convert radians to degrees, multiply by 59.2578.

## A COST-ESTIMATING WORKSHEET

Suppose you have a machine shop that does custom fabrication, and specializes in turning out finished circles from sheet steel. Each time someone orders your product or asks you for a quote, you must calculate the material required, labor required, and tax on the materials. The arithmetic functions make it easy to handle this job with a VisiCalc worksheet.

Figure 2-36 shows the cost-estimating worksheet you will enter in this lesson. It includes the following variables:

- Diameter — the diameter of the finished circle.
- Quantity — the number of circles ordered.
- Wastage — the amount of raw material lost to waste, expressed as a percentage of the finished area.
- Hourly Rate — the wage you must pay to produce finished goods.
- Labor Factor — the average number of labor hours (or fraction thereof) required per square foot of finished material.
- Stock Cost — your cost per unit of raw material.
- Tax Rate — the sales tax rate.
- Labor Margin — markup on Labor Cost.
- Material Margin — markup on Material Cost.
- Finished Area — the total area of all finished circles, calculated by multiplying Quantity, Pi, and the square of half of Diameter.
- Raw Area — the amount of raw material required, calculated by adding Wastage to Finished Area.
- Labor Cost — your cost of labor, calculated by multiplying Labor Factor, Finished Area, and Hourly Rate.
- Material Cost — your cost of materials, calculated by multiplying Raw Area and Stock Cost.
- Labor Charge — the amount you charge for labor, calculated by multiplying Labor Cost and Labor Margin.
- Material Charge — the amount you charge for materials, calculated by multiplying Material Cost and Material Margin.
- Tax — the sales tax, calculated by multiplying Tax Rate and Material Charge.

Diameter	3.25
Quantity	3
Wastage	.10
Hourly Rate	9.75
Labor Factor	.75
Stock Cost	3.85
Tax Rate	.07
Labor Margin	1.00
Material Margin	.50
Finished Area	24.89
Raw Area	27.38
Labor Cost	181.99
Material Cost	105.40
-----	
Total Cost	287.39
Labor Charge	363.98
Material Charge	158.10
Tax	11.07
=====	
Total Charge	533.14

010-015

**Figure 2-36. First Cost-Estimating Worksheet**

Clear the worksheet and type the following:

```

/GCC15 RETURN
/GF$
Diameter ↓ Quantity ↓ ↓ Wastage ↓ Hourly Rate ↓
Labor Factor ↓ Stock Cost ↓ Tax Rate ↓ ↓
Labor Margin ↓ Material Margin ↓ ↓ Finished Area
↓ Raw Area ↓ ↓ Labor Cost ↓ Material Cost ↓
/ - - → / - - ← ↓
Total Cost ↓ ↓ Labor Charge ↓
Material Charge ↓ Tax ↓
/ - = → / - = ← ↓
Total Charge RETURN
    
```

These are the labels for the worksheet. Compare to Figure 2-36.

To enter the numbers and formulas, type the following:

```

> B1 RETURN
3.25 ↓ /FI3 ↓ ↓ .1 ↓ 9.75 ↓
.75 ↓ 3.85 ↓ .07 ↓ ↓ 1 ↓ .5 ↓ ↓
+ B2* @PI*(B1/2^2) ↓
+ ↑ *(1 + B4) ↓ ↓
+ B5*B6*B13 ↓
+ B7*B14 ↓ ↓
+ ↑ ↑ ↑ + ↑ ↑ RETURN ↓ ↓
+ ↑ ↑ ↑ ↑ ↑ *(1 + B10) ↓
    
```

+ ↑↑↑↑↑ \* (1+B11) ↓  
+ B8\* ↑ RETURN ↓↓  
@SUM( ↑↑↑↑ . ↑ ) RETURN

The screen should look like Figure 2-37.

	A	B	C	D	E	F	G
Labor factor		1.75					
Body Cost		3.85					
Tax Rate		.07					
Labor Margin		1.00					
Material Margin		.50					
Finished Area		24.89					
Total Area		27.38					
Labor Cost		181.99					
Material Cost		105.40					
Total Cost		287.39					
Labor Charge		363.98					
Material Charge		158.16					
Tax		11.07					
Total Charge		533.14					

010-023/P

Figure 2-37. First Cost-Estimating Worksheet

Total Charge should be \$533.14. The Pi function is used to calculate the area of the circle; you'll use a few more arithmetic functions in a moment.

As long as the variables remain unchanged, estimating is simply a matter of entering the diameter and quantity of circles to be fabricated. You can add variables such as a Lookup function to handle stock that varies in cost, or use other formulas to calculate the areas of other shapes.

If you hide all entries except the customer-supplied information (diameter, quantity, etc.) and charges (labor, materials, and tax), you could use such a worksheet in the office or on the shop floor for on-the-spot quotes.

What if, once a sheet of material is cut, it cannot be reused? You must charge for the material by the sheet, not by the unit of area. That means you must know how many whole sheets the job requires. Any fraction of a sheet requires an additional sheet. You can change the worksheet to calculate this by using the Modulo Arithmetic (@MOD) and Integer Part (@INT) functions.

### Modulo Arithmetic

The Modulo Arithmetic function (@MOD) returns the remainder left after dividing the first argument as many times as possible by the second. For example:

@MOD(5,4)	= 1
@MOD(12,5)	= 2
@MOD(7,2)	= 1
@MOD(45.34,15)	= .34
@MOD(20,5)	= 0

As the last example shows, if the first argument is an even multiple of the second, the value of the function is 0.

### Integer Arithmetic

The Integer Part function (@INT) returns the integer part of its argument, ignoring any fractional part. The argument is truncated, not rounded. For example:

@INT(1.999)	= 1
@INT(9/2)	= 4
@INT(1/2)	= 0

As the last example shows, the integer part of a fraction less than 1 is 0.

You can use a combination of the Modulo Arithmetic and Integer Part functions to calculate how many sheets of material are required. A description of the calculation in words:

If the Raw Area divides evenly into the Sheet Size, the number of Sheets Required is the result of that division, otherwise it is one more than the result of that division.

A formula that contains an If function, an Integer Part function, and a Modulo Arithmetic function performs that calculation. You must add entries for Sheet Size and Sheets Required, and change Stock Cost from a per-unit-of-area cost to a per-sheet cost. Type the following:

```
> A5 RETURN
/IR
Sheet Size → /F150 ↓↓↓
192.5 RETURN
> A16 RETURN
/IR
Sheets Needed → /FI
@IF(@MOD(↑,B5)=0,
@INT(↑/B5),
@INT(↑/B5)+1) ↓↓↓
+ ↑↑↑ *B8 RETURN
```

The screen should look like Figure 2-38. The number of sheets needed, at B16, is 1. Total Charge has increased to 672.94.

Labor Margin	1.00
Material Margin	.50
Finished Area	24.00
Raw Area	27.30
Sheets Needed	1
Labor Cost	181.99
Material Cost	192.50
Total Cost	374.49
Labor Charge	363.98
Material Charge	288.75
Total Charge	652.73

010-024/P

**Figure 2-38. Second Cost-Estimating Worksheet**

Compare the formula for Sheets Needed (the three lines following Sheets Needed in the entry instructions) with the text description of the calculation. The two sets of three lines say the same thing, line-for-line.

Move the cursor to B2, and type **6 RETURN**. Sheets Needed increases to 2, because Raw Area is 54.75 and Sheet Size is 50. Total Charge increases to 1345.88.

Save the worksheet by typing **/SSCOST.ESTIMATE RETURN**.

The cost-estimating worksheet demonstrates how the arithmetic functions can be used, singly and in combination, to perform almost any calculation you need. The VisiCalc program includes many other functions. All the functions are described under "Functions" in the Reference.

## DRAWING BAR GRAPHS

An option of the Format command lets you represent values on the worksheet with a series of asterisks whose length is proportional to the value. A value of 1, for example, displays 1 asterisk; a value of 2 displays 2 asterisks, etc.

Try it. Clear the worksheet, and type **/F\*** (the Graph option of the Format command). Now type **5 RETURN**; A1 reads \*\*\*\*\*. Type **1 RETURN**, then **8 RETURN**, then **15 RETURN**. If the value is greater than the column width, the column fills with asterisks.

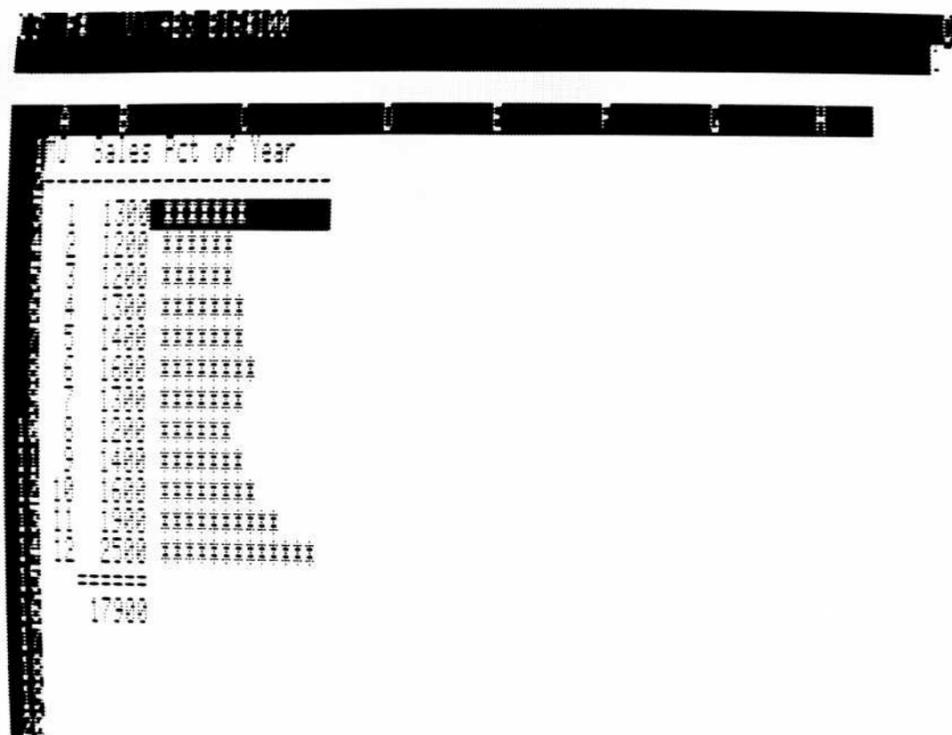
For an example of how to draw a graph, type the following:

```

/CAY
/GCC3 RETURN
MO → /FR
/GCC6 RETURN
Sales →
/GCC15 RETURN
" Pct of Year ← ← ↓
/— — RETURN
/R RETURN → . → → RETURN ↓
1 ↓ + ↑ +1 RETURN
/R RETURN ↓ . A14 RETURN R ↑ →
1300 ↓ 1200 ↓ 1200 ↓ 1300 ↓ 1400 ↓ 1600 ↓
1300 ↓ 1200 ↓ 1400 ↓ 1600 ↓ 1900 ↓ 2500 ↓
/— = ↓
@SUM(B3. ↑ ) RETURN
>C3 RETURN
/F*
+ ← /B16*100 RETURN
/R RETURN ↓ . C14 RETURN RN

```

This gives you a quick look at your yearly sales. The screen should look like Figure 2-39.



010-025/P

Figure 2-39. Graph of Monthly Sales

## SUMMARY

This concludes the Tutorial. Because its purpose is to teach basic use of the VisiCalc program, the Tutorial doesn't cover all details. But you have used all the key features. Now that you have completed the lessons, you can use three additional sources for information about the program: the Help command, the *VisiCalc Pocket Reference*, and the Reference chapter of this manual.

The Help command gives a brief description of each command, function, and feature. You can use it any time by typing a question mark (?); the information is usually sufficient to refresh your memory about a detail you've forgotten, or to give you a quick introduction to a feature you haven't used.

The *VisiCalc Pocket Reference* gives a brief summary of all the commands, functions, and features. Like the Help command, it aids your memory and gives a quick picture of a feature you haven't used.

The Reference chapter, however, gives a complete description of each command, function, or feature. It includes illustrations of the basic VisiCalc features (such as moving or replicating), examples of most commands and functions, and summaries of all options. To learn the details not covered in the Tutorial — all the options of the Attribute command, for example — turn to the appropriate topic in the Reference. The topics are listed alphabetically to make it easy for you to find what you're looking for.

You're ready to build VisiCalc applications that give you access to information that might otherwise take days or weeks to obtain or, in some instances, simply not be available. The examples in the Tutorial should give you some ideas; you'll find that after you build a few applications, ideas will come quicker than you can implement them.



## CHAPTER 3

### THE VISICALC®

### COMMAND REFERENCE

Introduction	3-2
Attribute Command - /A	3-5
Blank Command - /B	3-20
Clear Command - /C	3-21
Delete Command - /D	3-23
Edit Command - /E or Control-E	3-27
File Names	3-29
Prefixes and Pathnames	3-30
Subdirectories	3-31
Scrolling	3-31
Format Command - /F	3-33
The Functions - @	3-42
Global Command - /G	3-62
Go To Command - >	3-70
The Help Facility - ?	3-70
Insert Command - /I	3-72
Keystroke Memory Command - /K	3-76
Labels	3-87
Move Command - /M	3-90
Print Command - /P	3-96
Repeating Label Command - /-	3-104
Replicate Command - /R	3-106
The Screen	3-123
Storage Command - /S	3-127
Titles Command - /T	3-138
Values	3-140
Version Command - /V	3-148
Window Command - /W	3-149

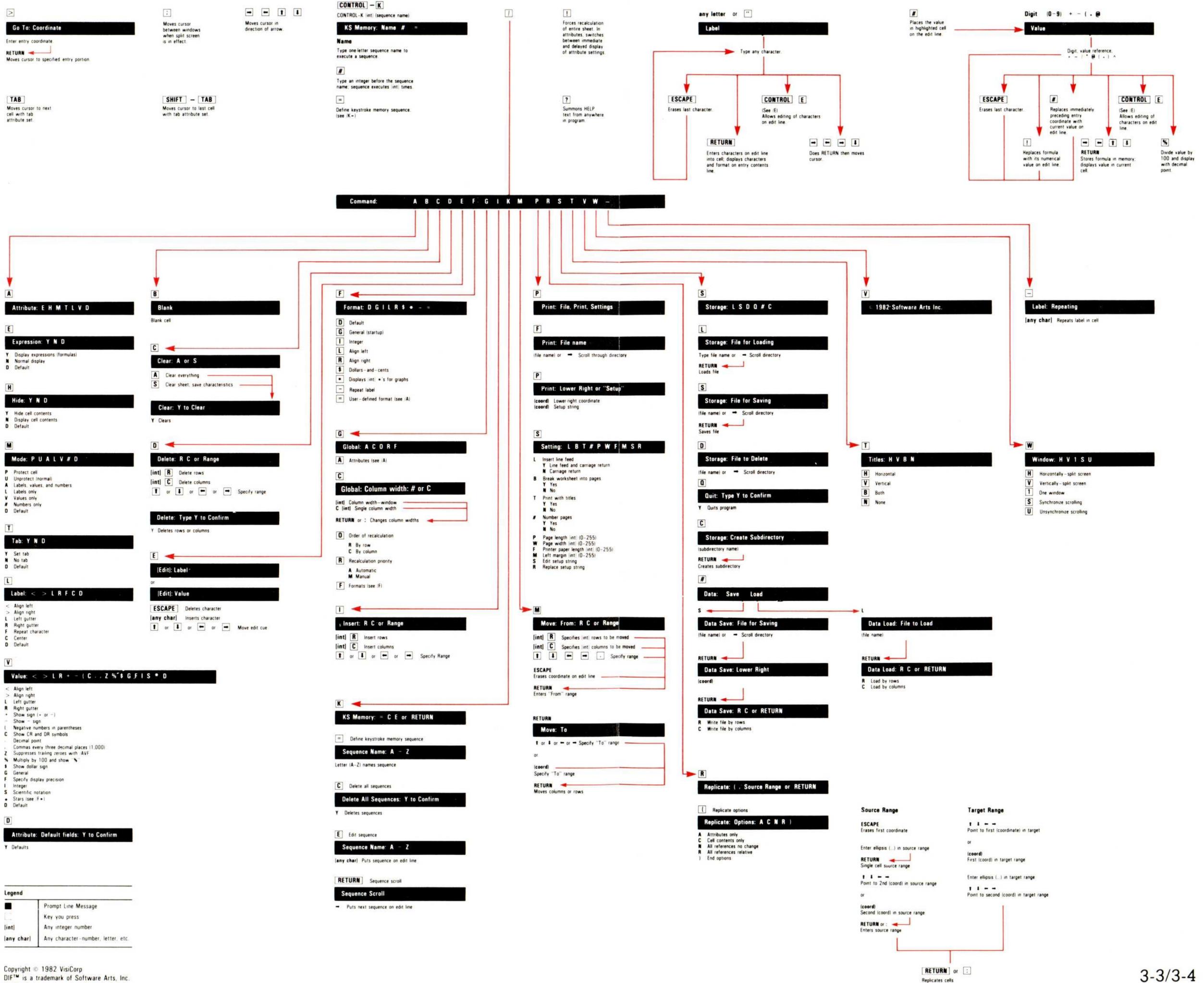
## **INTRODUCTION**

Chapter 3, the VisiCalc Command Reference, is a detailed alphabetical listing of each VisiCalc command and important topic. It answers any specific question you have about the VisiCalc program. While the Commands are listed alphabetically, the Command options are presented in the order they appear on the prompt line of the VisiCalc screen. Examples are used extensively to give the exact keystrokes needed to execute the command.

Before you use the Reference section, turn to "The VisiCalc Command Chart" and the section on "The Screen." The VisiCalc Command Chart shows all VisiCalc commands and their options. Each command description in the Reference begins with a simple diagram that shows all of its options. The section on "The Screen" introduces you to the key elements of the VisiCalc® Advanced Version screen.

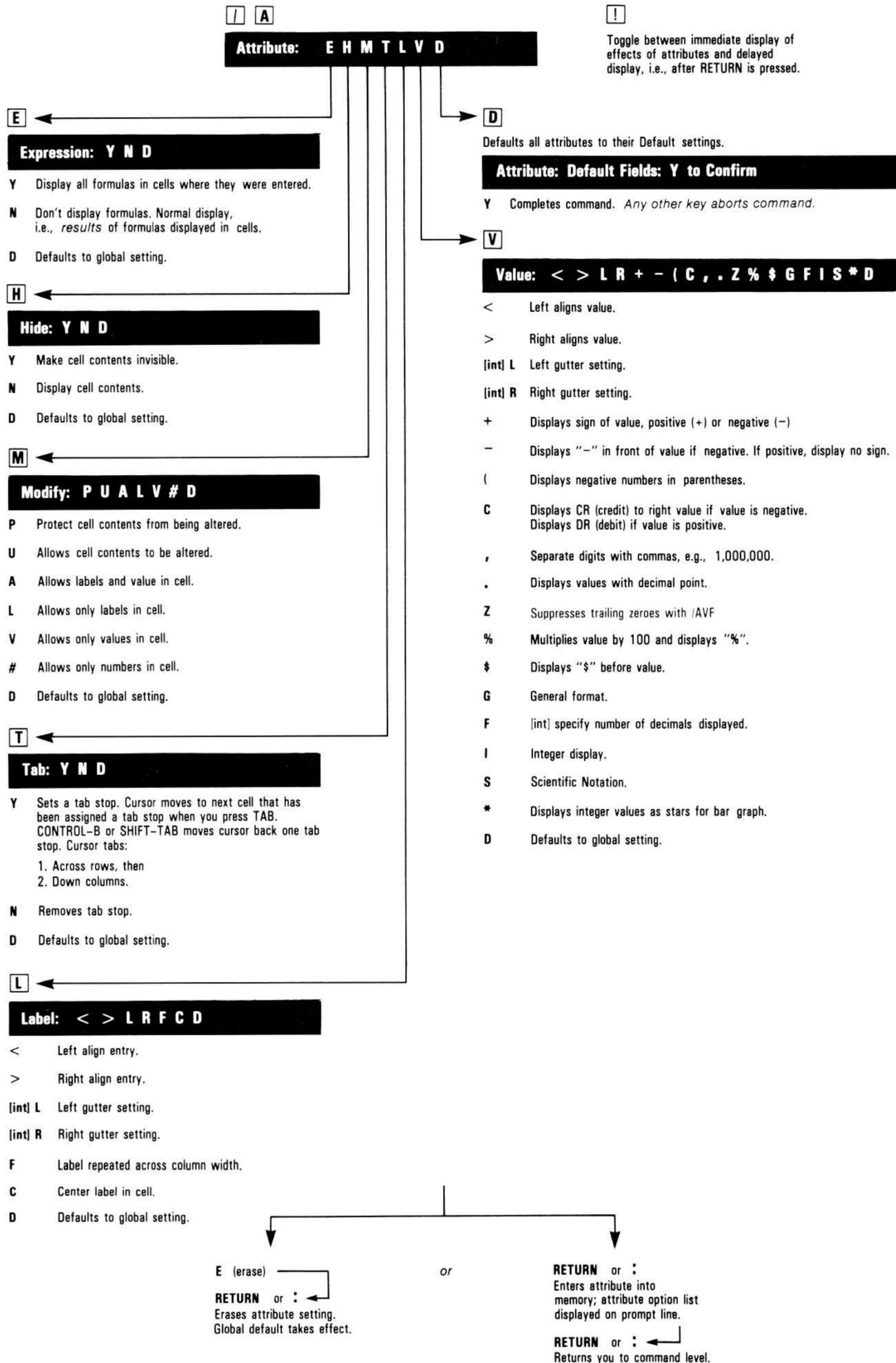
This Reference can be used in conjunction with the Tutorial or at any time you need to find a specific fact or detail about the VisiCalc program.

# VisiCalc® Advanced Version Command Structure Chart





# ATTRIBUTE COMMAND- /A

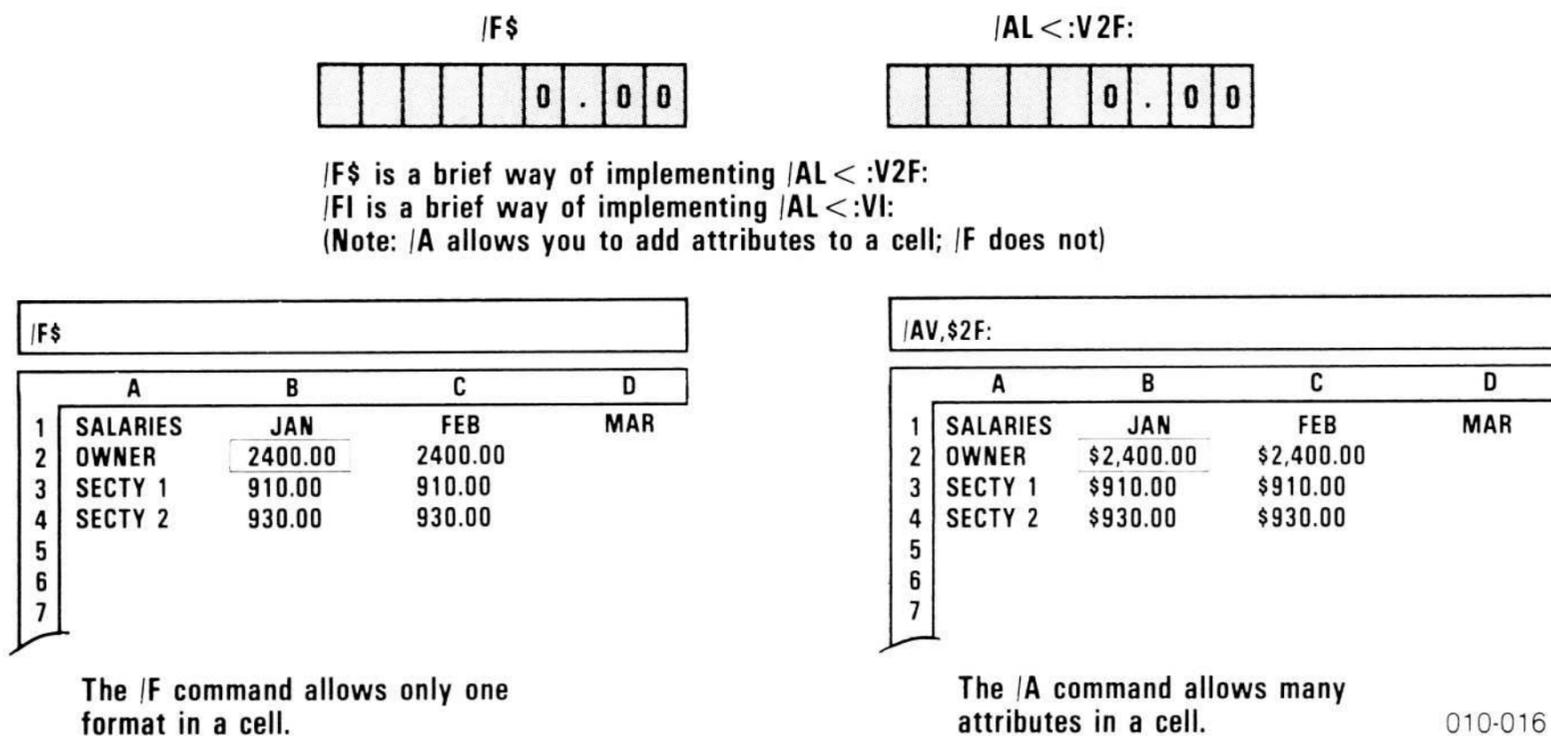


The Attribute Command sets characteristics for individual cells. To set attributes in a cell or to see which attributes are set, move the cursor to that cell, and type /A. When the cursor is in a cell that has attributes set, /A appears on the entry line.

The command has seven options:

Option	What It Does
E Expression format	Displays formulas in cells.
H Hide	Makes contents of a cell invisible.
M Mode	Defines whether or not data can be entered in a cell, and what kind of data (labels, values or numbers) can be entered.
T Tab	Sets tab in cell; cursor moves to next tab when the TAB key is pressed. CONTROL-B moves cursor to previous tab.
L Label	Sets label display characteristics.
V Value	Sets value display characteristics.
D Default	Sets all attributes to their current global defaults.

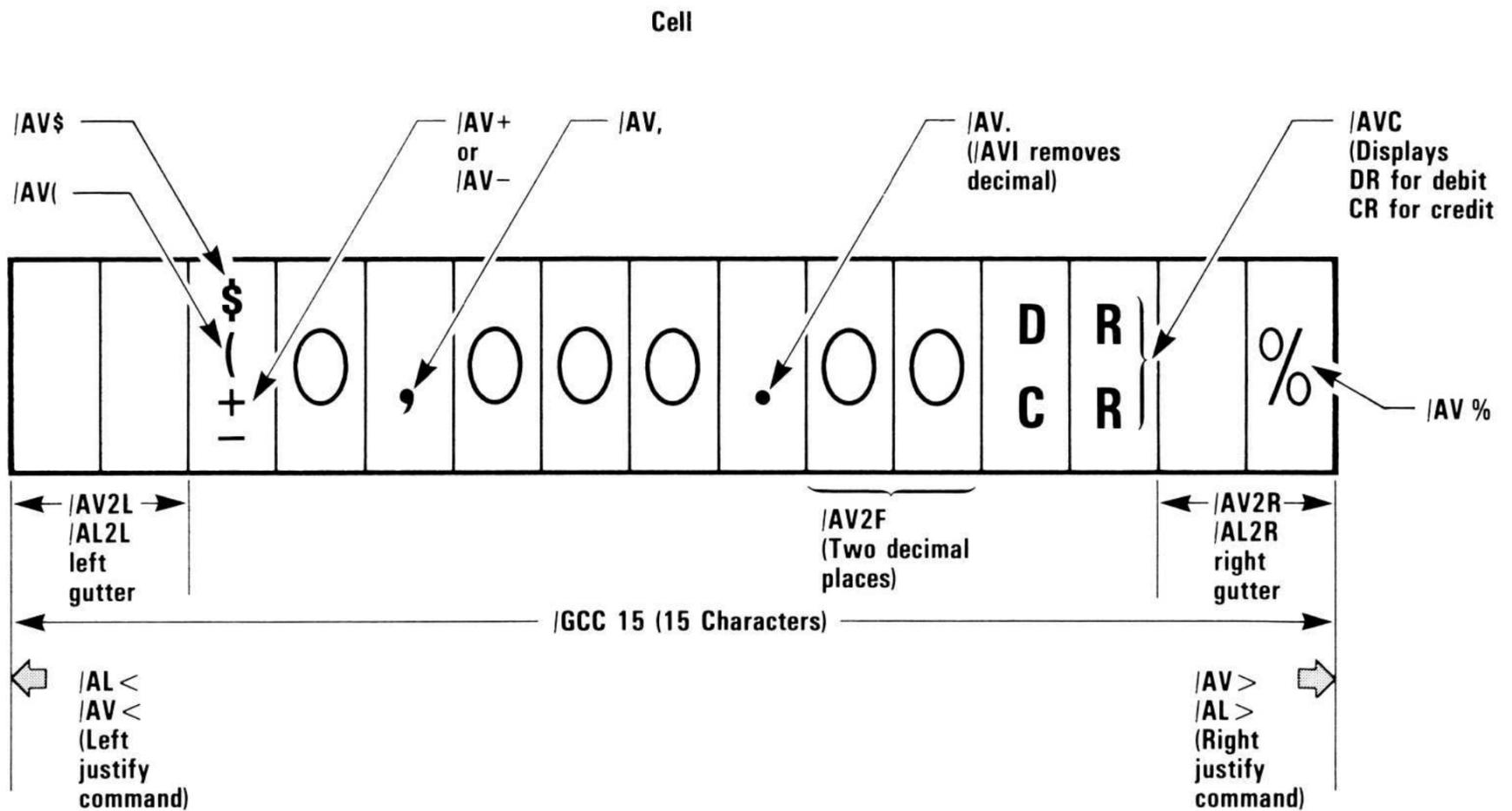
Some attributes control the cell display in much the same way as the Format command. In fact, the VisiCalc program actually converts formats into attributes, as Figure 3-1 demonstrates.



010-016

Figure 3-1. Comparing the Format and Attribute Commands

The Attribute command allows you more detailed control over cell displays than the Format command. Several attributes may be set in a cell; only one Format may be set in a cell. Figure 3-2 shows all the display characteristics controlled by the Attribute command.



**Notes:**

- /AHY Hides local cell
- /AEY Displays formula
- /ALD Defaults label setting
- /AVD Defaults value settings
- /AVS Displays in scientific notation

010-017

**Figure 3-2. Attribute Settings**

To see what attributes are in effect in a cell, move the cursor to that cell and type **/A**. The attributes are listed on the entry line. Cell formats set with the Format command (**/F**) are converted into attributes. For example, if you set a cell to integer format (with **/FI**) and type **/A** while still in the cell, **/AL < ; VI:** appears on the Entry Line. The cell displays a value as an integer or a label as left-aligned. Any attribute setting which makes a previous setting impossible replaces the previous setting.

Attributes may be set globally for a window with the /GA option of the Global command. Setting an attribute in a single cell—called a *local* setting—overrides any global setting for the same attribute. Note that the other five attribute settings are still controlled by the global setting.

When you save a worksheet, all local and global attributes are saved. Attributes do not affect values and labels in memory, only how they are displayed. Full precision values are always used in calculations.

Attributes not set globally or locally retain their program startup settings. The program startup settings are listed in Figure 3-3.

Figure 3-3 also shows the relationship between global and local attributes.

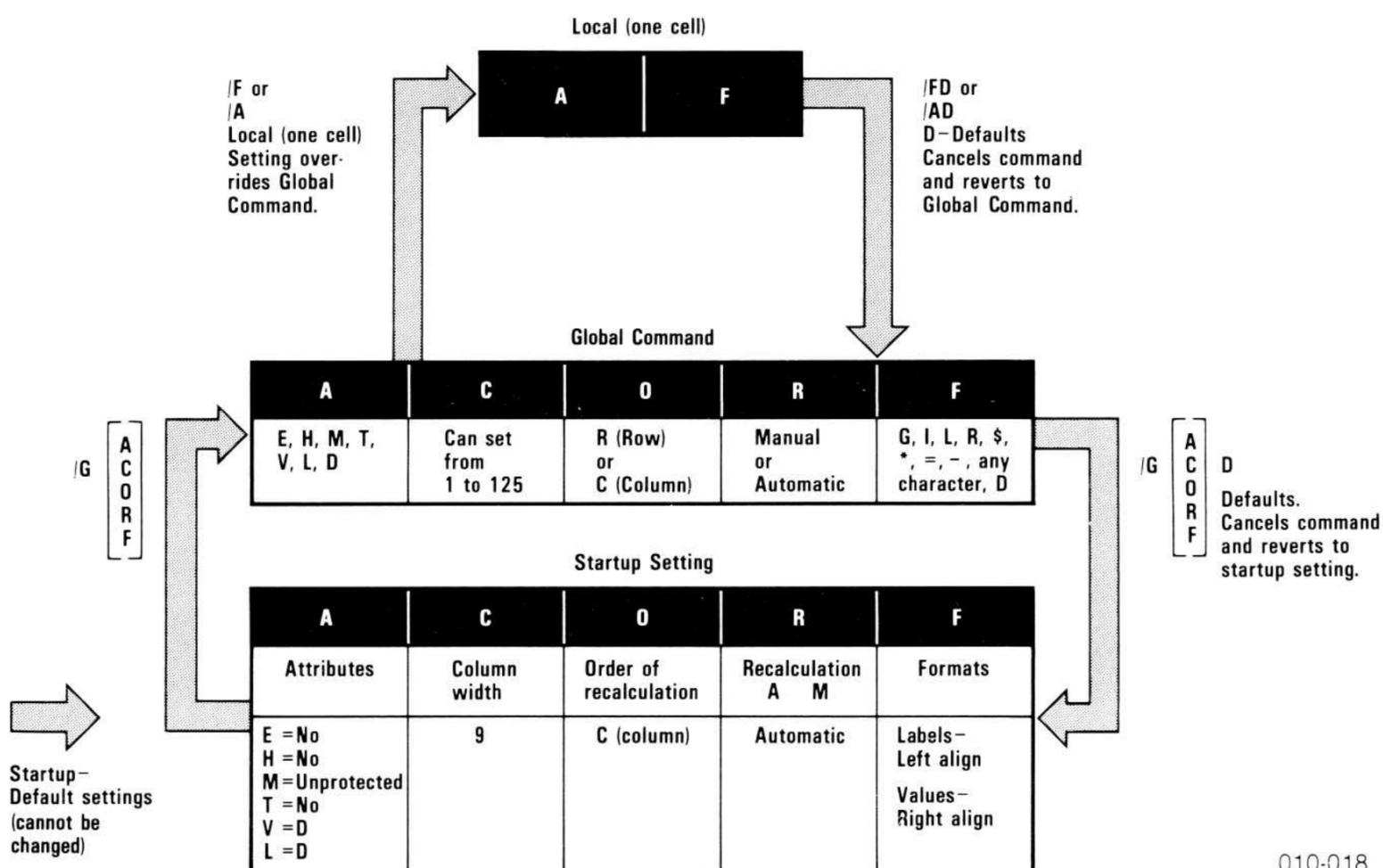


Figure 3-3. Local, Global, and Startup Settings

When you are entering an option from the Attribute command, the current setting of that option is displayed on the edit line. If you are setting attributes for one cell (with /A), a "!" will also appear on the edit line. Attributes take effect as you specify them when "!" is displayed. This allows you to experiment with different attribute settings. Attributes are not entered into memory until you press RETURN. Pressing ESCAPE before pressing RETURN cancels the latest setting. If you want to keep attributes from taking effect immediately, type !. The "!" will disappear from the edit line; an attribute setting will appear on the edit line as you specify it, but will not take effect until you press RETURN.

When you are setting global attributes (with /GA), the "!" does not appear on the edit line at first and an attribute does not take effect until you press RETURN. (This is useful because large worksheets with many entries may take up to several seconds to re-format each time an attribute is specified). If you type !, a "!" will appear on the edit line and attributes will take effect immediately.

A "!" appears as the last item on the prompt line for each attribute. This is to remind you that you can switch between immediate re-formatting and delayed re-formatting. For instance, if you type /AE, the prompt line will read Expression: Y N D !. Y, N, and D are the settings for the /AE option; the "!" is only a reminder, not a setting.

## Deleting Attributes

To delete cell attributes:

1. Place the cursor in the cell, and type /A.
2. Select the attribute you want to change (E, H, M, T, L, V, D).
3. Then type E, for erase. Type the specific setting you want to delete.

The following example shows how to delete an attribute setting.

Example:

<b>Type:</b>	<b>Result:</b>
1. /CAY	The worksheet clears, and the cursor moves to A1.
2. /AV., +RETURN RETURN	Values in cell A1 will be separated every three digits by commas, will be displayed with a decimal point, and will be preceded by their arithmetic sign, + or -.
3. /AVE	Edit Line: V+,.,:[ERASE] Program waiting for a specific setting to be selected for deletion.
4. +RETURN RETURN	The + setting is erased. Edit Line: V,.,:[ERASE]
5. /AHY RETURN RETURN	The Hide attribute has been set; contents of A1 will be invisible.
6. /AHE	Edit Line: HY:[ERASE] Prompt Line: HIDE: Y N D!
7. Y	The Y (Yes) setting is deleted. Edit Line: HD:[ERASE] Prompt Line: HIDE: Y N D! You can enter a new setting now for the Hide option.

## The Expression Format Option — /AE

The Expression Format option displays the expression (formula) in the cell. (Normally, the calculated result of the formula appears in a cell.) The Expression Format option is useful to examine the logic of a worksheet, because it lets you see which cells are referenced by each formula in a worksheet.

When you type **/AE**, the prompt line displays **EXPRESSION: Y N D!**. (Remember, the **!** is simply a reminder that you can switch between immediate and delayed re-formatting.) The three options are:

- Y** Yes, display the formula in cell instead of the result.
- N** Normal cell display, i.e., the result of the formula.
- D** Set this cell to the global default last set with **/GAE**.

The **/AE** option displays in the cell what you would see on the entry line if you moved the cursor to that cell. The following example shows how the **/AE** option works.

Example:

<b>Type:</b>	<b>Result:</b>
1. <b>/CAY</b>	The worksheet clears, and the cursor moves to <b>A1</b> .
2. <b>1 RETURN</b>	Entry Line: <b>A1 (V) 1</b>
3. <b>↓2 RETURN</b>	Entry Line: <b>A2 (V) 2</b>
4. <b>↑→ +A1/A2 RETURN</b>	Entry Line: <b>B1 (V) +A1/A2</b> Cell <b>B1</b> displays <b>.5</b> .
5. <b>/A</b>	Prompt Line: <b>Attribute: E H M T L V D</b>
6. <b>E</b>	Prompt Line: <b>Expression: Y N D!</b> Edit Line: <b>ED:!</b>
7. <b>Y</b>	Edit Line: <b>EY:!</b>
8. <b>RETURN RETURN</b>	Cell <b>B1</b> displays <b>+A1/A2</b>  The result of the formula in this cell is not displayed, since the cell is in expression format.

Setting this option globally, with /GAE, allows you to see every formula on the worksheet. Figure 3-4 shows a worksheet displayed in expression format.

Midwest Hardware S 327 E. Mountain Av Berthoud, CO 80513		upply enue	No: 201 Date: 10/16/83
Bill to: ----- Ranch Supply Co. 2025 S. Main Longmont, CO 80310		Ship to: ----- SAME	
Terms: Net 30	Ship via:Truck	Resale #:	1234567
			Cust. #:
			200
Item Number	Desc.	Quantity Ordered	Quantity Shipped
			Retail Price
			Disc. Price
			Total Price
			Back Ordered
20416	Gloves	100	10
			@LOOKUP(E20,A4...A7)
			+I20-(I20*(@LOOKUP(G20,C3...C10)/100)
			+J20*H20
			+G20-H20
20446	Glue	36	36
			@LOOKUP(E21,A4...A7)
			+I21-(I21*(@LOOKUP(G21,C3...C10)/100)
			+J21*H21
			+G21-H21
21329	Hammer	100	100
			@LOOKUP(E22,A4...A7)
			+I22-(I22*(@LOOKUP(G22,C3...C10)/100)
			+J22*H22
			+G22-H22
21377	Nails	50	50
			@LOOKUP(E23,A4...A7)
			+I23-(I23*(@LOOKUP(G23,C3...C10)/100)
			+J23*H23
			+G23-H23
		Subtotal	
		@SUM(K20...K28)	
		Shipping	
		@MAX(B16*K29,B17)	
		Tax	
		@IF(J15>0,0,B15*K29)	
		Total	
		@SUM(K29...K32)	
		Amt. Rcv.	
		-----	
		Amt. Due	
		+K33-K34	

010-009

**Figure 3-4. Worksheet Displayed in Expression Format**

To print a worksheet in expression format, follow these steps:

1. Type /GAE Y RETURN RETURN to put the entire worksheet in global display format.
2. If any cells were set locally to non-expression format (with /AEN), reset them to the global default (with /AED).
3. Make sure your columns are wide enough to display every term in every formula on the worksheet.
4. Issue the normal print command for your printer.

### **The Hide Option—/AH**

The Hide option makes the contents of a cell invisible.

When you type /AH, Hide: Y N D! appears on the prompt line. The three option settings are:

- Y Yes, hide the contents of this cell.
- N No, don't hide the contents of this cell.
- D Set this cell to the default last set with /GAH

The /AH option overrides /AV (Value) and /AL (Label). The /AE option, which displays formulas and expressions in cells, overrides /AH.

### **The Mode Option—/AM**

The Mode option controls whether or not data can be entered in a cell, and what kind of data can be entered (labels, values or numbers).

When you type /AM, this prompt appears: MODE: P U A L V # D!. The settings for this attribute are:

- P Protects the contents of a cell from being changed; no entry is allowed in the cell.
- U Allows cell contents to be changed.
- A Allows you to enter all data types—numbers, formulas, and labels.
- L Allows you to enter only labels; numbers and formulas are not allowed.
- V Allows you to enter only numbers and formulas; labels are not allowed.
- # Allows you to enter only numbers; formulas and labels are not allowed.
- D Returns setting to global default last set with /GAM.

(The ! switches between immediate and delayed re-formatting.)

Figure 3-5 illustrates the various settings of the Mode option.

Settings	Cell will accept	Cell will not accept
/AMV (only values)	+B7/B24-A2 @SUM (A1...A10) 55.7	LABEL 1 "-GROSS-+@"
/AML (only labels)	LABEL 3 "-GROSS-+@"	+B14-A2 55.7
/AM# (only numbers)	24.5691 10	+B7/B24-A2 LABEL 3 @SUM (A1...A10)
/AMA (all)	55.7 @SUM (A1...A10) "-GROSS-+@" LABEL 1 24.5691	

010-019

**Figure 3-5. Mode Option Settings**

The Mode option can save you time entering values and labels. When you have the Mode option set to L (labels only) you do not need to begin a label whose first character is non-alphabetic with a quotation mark ("). For instance, if the label-only setting (/AML) is in effect in a cell, you can type the first character in the label -NET- without typing " first. When the first character is a command character—/, ;, >, [SPACE]—you must type the quotation mark (") first, however.

Likewise, if the Mode is set to /AMV (values only), you do not need to type + to begin a formula whose first character is a cell coordinate (A1/B1 - C1). If you point to a cell with the cursor to enter a value, however, you must type the + first.

**WARNING:** You can change attributes and formats in a protected (/AMP) cell, and you can replicate into a protected cell; /AMP protects only data entry. Replicating into a cell in which /AMP has been set will change that setting to whatever was replicated. This is also true where /AMA, /AML, /AMV or /AM# are in effect.

## The Tab Option—/AT

The Tab option specifies what cell the cursor will move to next when you press **TAB** or **CONTROL-B** or **SHIFT-TAB**. You can set tabs at the places where data is to be entered, avoiding unnecessary cursor movement to find the next entry location.

The following example demonstrates the Tab option.

Example:

Type:	Result:
1. /CAY	The worksheet clears and the cursor moves to A1.
2. /A	Prompt Line: ATTRIBUTE: E H M T L V D
3. T	Prompt Line: TAB: Y N D! Edit Line: TD:!
4. Y RETURN RETURN	Edit Line: TY:! A tab is now set at A1.
5. > A5 RETURN	Cursor moves to A5.
6. /ATY RETURN RETURN	Tab is now set at A5.
7. > A10 RETURN	Cursor moves to A10.
8. /ATY RETURN RETURN	Tab is now set at A10.
9. > C1	Cursor moves to C1.
10. /ATY RETURN RETURN	Tab is now set at C1.
11. > A1	Cursor moves to A1.
12. TAB	Cursor moves to C1. Tabbing is always done across rows first, then down columns.
13. TAB	Cursor moves to A5.
14. TAB	Cursor moves to A10.
15. CONTROL-B	Cursor moves back to A5. Typing CONTROL-B moves the cursor to the last tab, in the reverse order.

Pressing **TAB** at the end of a command executes the command and moves the cursor to the next tab-set cell. Pressing **CONTROL-B** or **SHIFT-TAB** (the **TAB** key while holding down the **SHIFT** key) executes the command and moves the cursor to the last cell with a tab set.

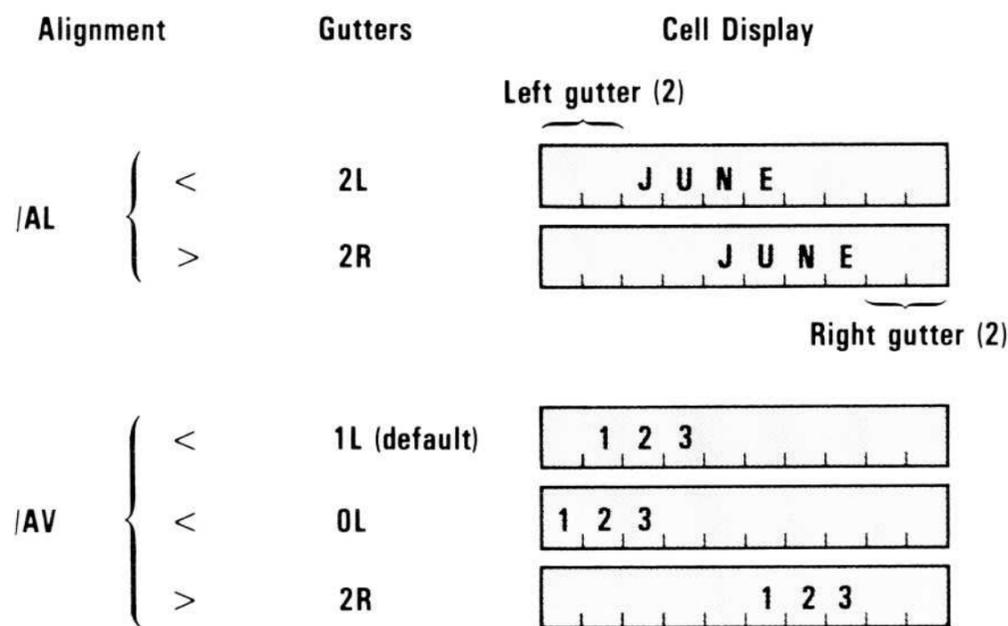
### The Label Option—/AL

The Label option controls how a label is displayed. When you type /AL, the prompt on the prompt line is : < > LRFCD!. The seven options are:

- < Left-aligns the label between gutters.
- > Right-aligns the label between gutters.
- L Determines the number of character spaces between the left edge of the cell and the beginning of the label (left gutter).
- R Determines the number of character spaces between the right edge of the cell and the end of the label (right gutter).
- F Repeats the label across the column between gutters; similar to the Repeating Label command (/—).
- C Centers the label between gutters.
- D Returns the setting to the current global default.

(The ! switches between immediate and delayed re-formatting.)

Figure 3-6 shows how gutters and alignment interact.



010-020

**Figure 3-6. Gutters and Alignment in Cell Displays**

The /ALC option centers a label between the the left and right gutters. Thus, if the left gutter is larger than the right gutter, the label is not displayed in the center of the column. To insure that a label will be exactly centered: (1) do not set any gutters, or (2) set the left and right gutters to the same value. Labels with an odd number of characters cannot be exactly centered; the extra character space is to the right of center.

The default setting for label gutters is 0R 0L, i.e., no gutters.

## **The Value Option—/AV**

The Value option sets value display characteristics, such as aligning values right or left, putting decimal points in values, displaying the sign (+ or -) of a value, and others shown below.

After typing /AV, the prompt line will show:

VALUE: <>LR + -[C ,.Z% GFIS\*D]

The settings for this option are:

- < Left-aligns a value between gutters.
- > Right-aligns a value between gutters.
- L Sets the number of character spaces between the left edge of a cell and the beginning of a value. Takes an integer argument. If no integer is supplied, 1 is assumed.
- R Sets the number of character spaces between the right edge of a cell and the end of a value. Takes an integer argument. If no integer is supplied, 0 is assumed.
- + Positive values are preceded with a plus sign (+), and negative values are preceded with a minus sign (-). 0 has no sign.
- Normal value display goes into effect; negative numbers are preceded with a minus sign; positive numbers and 0 have no sign. (Useful as local setting to suppress leading "+" or "-" when /AV+ is set globally.)
- ( Negative numbers are displayed in parentheses, and positive numbers and 0 are displayed with no sign nor parentheses.
- C Positive numbers are displayed with the Debit symbol (DR), and negative numbers are displayed with the Credit symbol (CR).

- , Commas appear every three digits in numbers with four or more digits to the left of the decimal; e.g., 1000 is displayed as 1,000 and - 1000 is displayed as - 1,000; does not affect numbers displayed in scientific notation.
- . The decimal point is displayed whether or not any digits are to the right of the decimal point. For example, 3 is displayed as "3."
- Z All trailing zeroes are suppressed in cells with /AVF or /AVS settings.
- % Values are multiplied by 100 and displayed with percent sign (%). .23, for example, appears as 23%.
- \$ Displays a value with the dollar sign. Unlike /F\$, displays the full decimal precision of the value; 2.345 appears as \$2.345.
- G General value display. Same as the initial program default for value displays.
- F Values are displayed in fixed format with the specified number of digits after the decimal point; trailing zeroes are displayed unless /AVZ is also in effect; this setting takes an integer argument, and if none is supplied, the general display takes effect.
- I Displays values in integer form. Overrides /AVZ.
- S Displays values in scientific notation. This setting overrides /AV. , /AV, etc.
- \* Displays values in graph format; same as /F\* (see "The Format Command" later in this chapter).
- D Reverts to current global setting; set with /GAV.

The best way to see if two settings work together, or to see how your display looks when two settings are in effect, is to *try it out yourself*. You can't harm your data, and you can always change the value settings back to their defaults.

The table below presents a few commonly-used formats and the attributes needed to obtain them.

<b>Attributes</b>	<b>Format</b>	<b>Sample Display</b>
/AV., +	Values separated by commas, with decimal point, preceded by + or - sign.	+ 1,000.
/AV2F\$	Two decimal places shown, including trailing zeroes; displayed with dollar sign.	\$1000.00
/AV(	Negative numbers enclosed in parentheses, accounting-statement style.	(1000)
/AVC	Positive numbers displayed with Debit symbol (DR); negative numbers displayed with Credit symbol (CR).	1000CR
/AV4F	Number displayed with four digits after decimal point.	1000.0000
/AV%	Number (.10) multiplied by 100 and displayed with percent sign (%).	10%

**Note:** Be aware that certain attribute settings cause values to be rounded for display. Only the displayed value is rounded; the program uses unrounded values for all calculations.

### **The Default Option—/AD**

The Default option causes a cell to be displayed according to the current global settings. (Global settings are specified with the /GA option of the /G command; see "The /GA Option" for more information.)

**BLANK COMMAND—/B**

| B

**Blank**

↑ ↓ → ← or RETURN

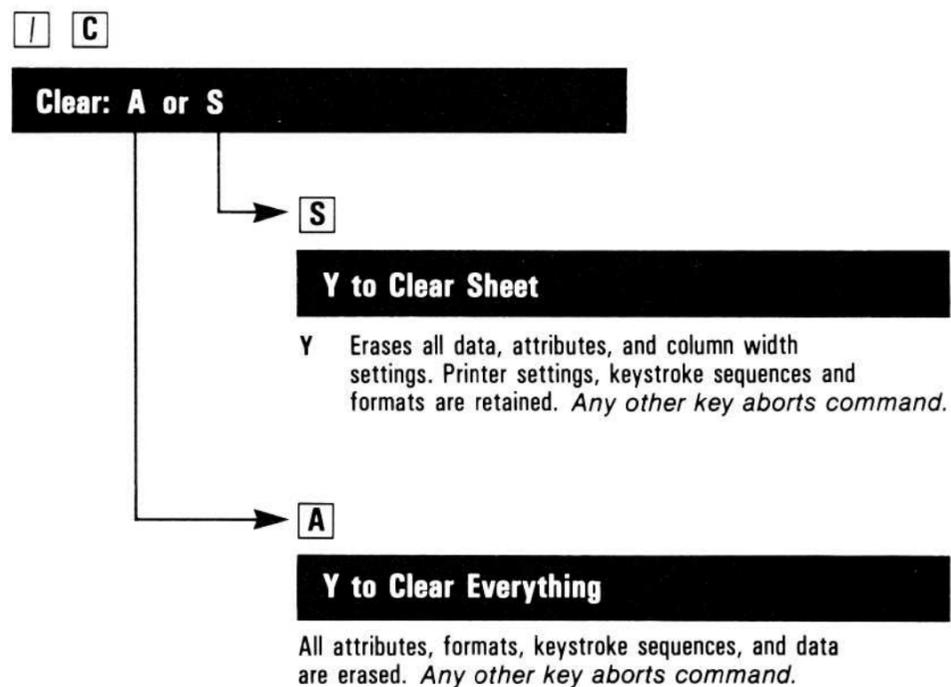
Erases contents of highlighted entry position, leaves format. *Any other key aborts command.*

The Blank command *irretrievably erases* any label or value in the cell where the cursor rests. It does not affect local formats or attributes, if they were specified. /B does not work in cells where /AMP is set.

Example:

Type:	Result:
1. Place the cursor on the entry to be erased.	
2. /	Prompt line: Command: ABC DEFGIKMPRSTVW—
3. B	Prompt line: Blank
4. RETURN or an arrow key.	The cell highlighted by the cursor clears.

## CLEAR COMMAND—/C



The Clear command *irretrievably erases* all data on the worksheet and returns the cursor to position A1. If you want to save the worksheet (with /SS), be sure to do it before issuing the Clear command. You should always use the Clear command before starting or loading a worksheet.

The command has two options:

- A Sets all worksheet characteristics to the same status as when the program is loaded.
- S Clears worksheet entries and attributes; formats, printer settings, and keystroke sequences are retained.

The VisiCalc program is too big to fit in current memory. /CS keeps whatever commands were entered most recently in current memory, while /CA loads the Startup commands only. If you used /R recently, clearing the screen with /CS keeps the Replicate command in memory; the VisiCalc program does not have to go out to the program diskette to get the Replicate command. If you clear the screen with /CA, however, the program has to go out to the program diskette to get the Replicate command next time you type /R.

## The Clear All Option—/CA

The /CA option clears:

- Labels and values
- Keystroke Memory sequences (/K =)
- Printer settings made with /PS
- Formats and attribute settings
- Column width settings
- Titles settings
- User-defined formats (/F =)

## The Clear Sheet Option—/CS

The /CS option clears:

- Labels and values
- Formats and attribute settings
- Column width settings
- Title settings

*/CS does not clear* the following worksheet characteristics:

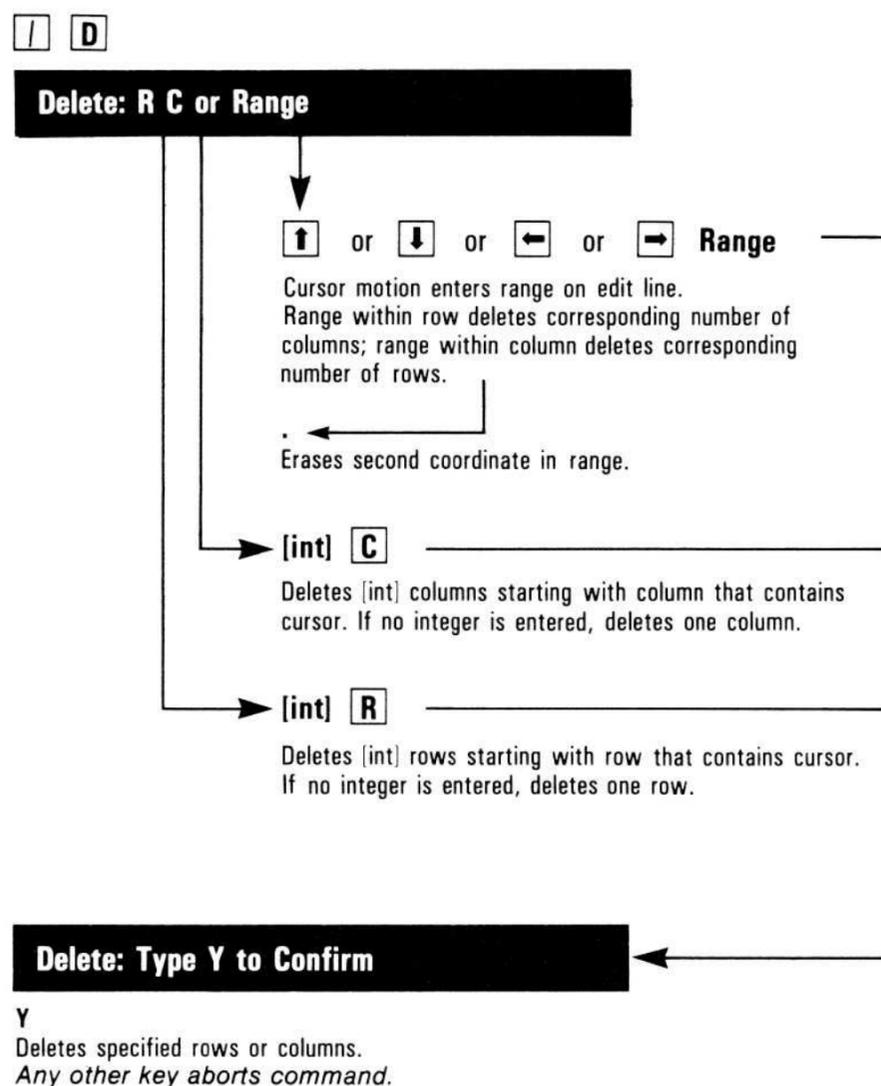
- Printer settings made with /PS
- Definitions of formats you designed yourself using /F =
- Keystroke Memory sequences

Example:

Type:	Result:
1. /C	Prompt Line: Clear: A or S.
2. A	Prompt Line: Type Y to confirm.
2. Y	The screen darkens for a moment, then reappears with no entries. The status area displays the copyright and version notices.

If you type anything other than Y, the Clear command is canceled.

## DELETE COMMAND—/D



The Delete command *irretrievably erases* rows or columns. If rows are deleted, all rows below the deleted rows move up and fill the deleted rows; if columns are deleted, all columns to the right move left to fill the deleted columns.

The VisiCalc program adjusts all references on the worksheet to correspond to the new coordinates that result when rows or columns are moved after the Delete command, then recalculates the worksheet. Cells that refer to a deleted row or column display `ERROR`.

After you type /D, the Prompt Line reads:

Delete: R C or Range

At this point, you can : (1) Type R or C, which specifies one row or column to be deleted, starting with the row or column where the cursor is, (2) type an integer that specifies how many rows or columns are to be deleted, starting with the row or column where the cursor is, then type R or C, or (3) enter a range of columns or rows, either by moving the cursor or typing the first and last coordinates. Type Y in response to the Y TO CONFIRM to delete the rows or columns; any other key aborts the command.

Figure 3-7 shows how several columns are deleted.

a) Before Deleting

	A	B	C	D	E	F	G	H
1	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG
2								
3	100	200	300	400	500	600	700	800
4								
5								
6								
7	With cursor on "FEB", type /D2C, meaning "delete two columns starting here." Type Y to confirm.							

b) After Deleting

	A	B	C	D	E	F	G	H
1	JAN	APR	MAY	JUN	JUL	AUG		
2	100	400	500	600	700	800		
3								
4								
5								
6								
7	"FEB" and "MAR" columns disappear. Columns to the right close the gap.							

010-021

**Figure 3-7. Deleting Columns**

To specify a range of columns or rows to be deleted, place the cursor in the first row or column you want to delete, type /D, move the cursor to the last row or column you want to delete, and press RETURN. The deleted range will be from the first row or column to the last row or column.

To delete columns, specify the range within any row. The ranges A1...D10, A100...D100, and A254...D254 delete columns A-D.

To delete rows, specify the range within any column. The ranges A1...A10, Z1...Z10, and BK1...BK10 delete rows 1-10.

Here is an example of how to use the Delete command.

Example:

- | <b>Type:</b>                                | <b>Result:</b>   |
|---|--|
| 1. <b>/CAY</b>                              | The worksheet clears and the cursor moves to <b>A1</b> . |
| 2. <b>1 → 2 → 3 →</b><br><b>1+C1 RETURN</b> | The screen should look like Figure 3-8.                  |



010-026/P

**Figure 3-8. A Reference Before Deleting a Column**

- |                         |  |
|-------------------------|--|
| 3. <b>&gt;B1 RETURN</b> | Cursor moves to <b>B1</b> .                      |
| 4. <b>/D</b>            | <b>Prompt Line:</b><br>Delete: R C or Range      |
| 5. <b>C</b>             | <b>Prompt Line:</b><br>Delete: Type Y to Confirm |

6. Y

The screen should look like Figure 3-9.



010-027/P

**Figure 3-9. Reference After Deleting a Column**

7. →

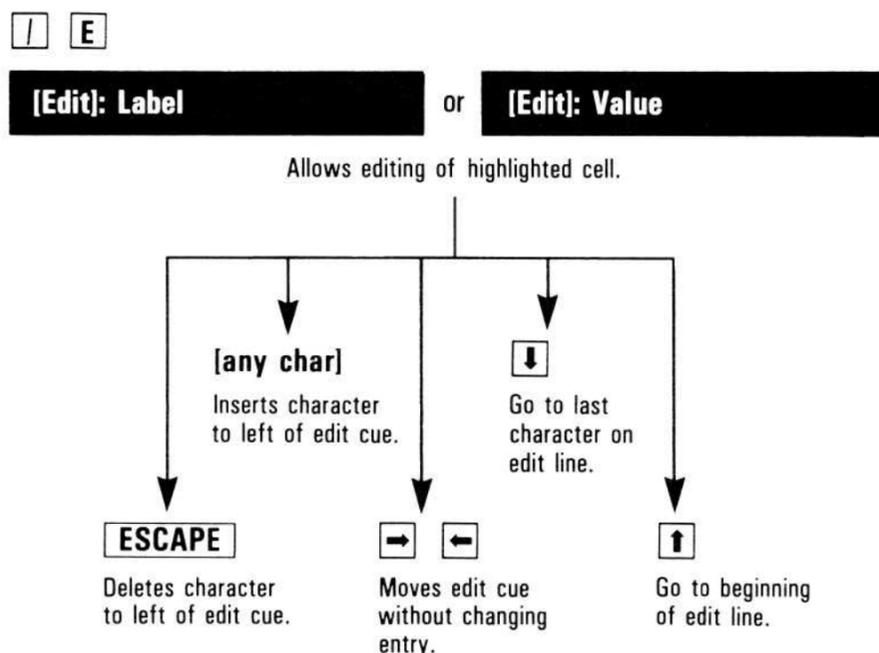
Entry Line: C1 (V) 1+B1

When column B was deleted, all columns to the right moved left to fill the gap. This formula has been changed from 1 + C1 to 1 + B1 so that it still refers to the value you entered at C1 in step 2.

**Note:** Use the Delete command with care. You cannot recover the information from a row or column after you have deleted it. The Insert command creates space for a row or column, but does not recover deleted information.

If you delete an entry that is referenced in a formula at another location, all locations that refer to the deleted entry display **ERROR**.

## EDIT COMMAND—/E OR CONTROL-E



The Edit command lets you change an entry without retyping the entire value or label. You can insert or delete characters anywhere in the entry; this is especially helpful when you must change a long or complex formula. Two forms of the Edit Command can be used: to edit something being typed on the edit line (use **CONTROL-E**); to edit an entry already written on the worksheet (use **/E**).

After you type **/E** or **CONTROL-E**, the entry to be edited is on the edit line. If you type **/E**, the edit cue is at the beginning of the edit line; if you type **CONTROL-E**, the edit cue is at the end of the edit line. You can move the edit cue along the edit line without affecting the entry. Typing **←** moves it left one character, **→** moves it right; **↑** moves it to the beginning of the line, and **↓** moves it to the end. Typing **ESCAPE** erases the character to the left of the edit cue; typing any other character inserts that character to the left of the edit cue.

Type RETURN to end the edit command, or type CONTROL-C to cancel it.

To edit the entry at the cursor location, type /E. To edit an entry being typed (not yet written on the worksheet), type CONTROL-E.

Example:

Type:	Result:
1. /CAY	The screen clears and the cursor moves to A1.
2. <b>mistek</b>	Prompt Line: Label Edit Line: mistek The edit cue follows the k.
3. <b>CONTROL-E</b>	Prompt Line: [Edit]: Label Edit Line: mistek The edit cue follows the k.
5. ←←	The edit cue highlights the a.
6. <b>ESCAPE</b>	Edit Line: mistak The edit cue still highlights the a.
7. ↓	Edit Line: mistak The edit cue follows the k.
8. <b>e RETURN</b>	Entry Line: A1 (L) mistake Prompt Line: clear Edit Line: clear A1: mistake

## **FILE NAMES**

Each time you save a worksheet to a storage device, such as a diskette, you create a file. Every file on a diskette must have a unique name, known as its file name, to differentiate it from all other files on the diskette. If you do not use subdirectories (see "The Storage Command" in the Reference and below), the file name is called the local name. If you use subdirectories, you can use the same local name once per subdirectory.

If you try to save a file under a local name that you've given to an earlier file, the contents of the more recent file will take the place of the contents of the earlier file, effectively erasing it. You should save to an existing file name only when you are updating the existing file.

A valid local name or subdirectory name can include up to 15 characters. The first character must be a letter. The rest of the name can include letters, numbers, and periods. No other characters, including the space character, are allowed in a local or subdirectory name.

Thus, FILETWO.VC and FILE2.DIF are both valid. Two examples of invalid names are 1980ACCOUNTS (begins with a number) and MYDISKETTE NO.TWO/PF (includes a space, a slash, and is longer than 15 characters).

Note the suffixes in the above local name examples—.VC, .DIF, and .PF. They represent the three types of files that VisiCalc can create. Their purpose is to help you and VisiCalc differentiate between the three file types.

We recommend that you append .VC to any standard VisiCalc local name (one saved with /SS), .DIF to any DIF format local name (one saved with /S#S), and .PF to any Print Format local name (one saved with /PF). See Part III, "The Storage Command" and "The Print Command" for further details.

## PREFIXES AND PATHNAMES

Whenever you format a diskette (see Appendix A), SOS (the Apple III Operating System) not only prepares the diskette to receive data, but also creates a directory known as the main or root directory. The directory stores the file names and file addresses of all the files on its disk.

Because the directory is a file, SOS gives it a name, the volume name. The formatting process creates the name /BLANK. With the Apple III Utilities Diskette, you can change the volume name on any diskette; i.e., /BUDGET, /MYDISK1, /SALES3, etc. In fact, it's a good idea to give each volume a unique name. See the *Apple III Owner's Guide* for details on how to rename a diskette directory.

Besides naming a directory and recognizing directory names, SOS also recognizes device names. Every device connected to the Apple III has a name. For example, the built-in disk drive is .D1. The other floppy drives are .D2, .D3, and .D4. If you are using the ProFile hard disk, the device name is .PROFILE. The device name for the Corvus hard disk is .C1. If you have connected a Silentype™ printer, SOS recognizes it as .PRINTER.

Either a device name or a volume name can serve as a prefix to the local name of a file. Note that a device name always begins with a period (.) and a volume name always begins with a slash (/). The slash also serves to separate either type of prefix from the local name. The prefix-slash-local name combination is known as a pathname.

If you want to locate a file on a drive other than the default drive, you must use the pathname. As an example, let's say that the built-in drive was the default drive and that drive 2 contained a diskette with the volume name /SALES and a local file named FORECAST.VC. To load FORECAST.VC, you would first type /SL, followed by either .D2/FORECAST.VC or /SALES/FORECAST.VC.

When you specify a device or volume with a prefix, SOS makes that prefix the default value. You won't have to specify a prefix again until you go to another device or volume.

## **SUBDIRECTORIES**

SOS also lets you create subdirectories with the /SC command (see "The Storage Command"). A subdirectory is a file that points to other files, even to other subdirectories. You don't have to store your files by subdirectories, but they are extremely handy for hierarchical structuring of data.

Once you have created a subdirectory, it becomes the default reference for all storage commands. To gain access to other files and subdirectories, you must name them as part of a pathname.

For example, let's say you have two subdirectories on a diskette, one named SUB1 and the other named SUB2. Further, each of these subdirectories contains two files, FILE1 and FILE2. If you are working with FILE1 in SUB1 and want to use FILE1 in SUB2, you would type /SL, the Load command, followed by .D1/SUB2/FILE1. This assumes that the diskette containing these files is in the built-in drive.

Note that after typing .D1/SUB2/, you can scroll through the local names in SUB2 (see "Scrolling" below).

Subdirectories cannot be deleted until you have first deleted all the files that they point to. When this is done, you must use the regular SOS DELETE command. /SD will not delete subdirectories. For a thorough discussion of subdirectories, see the *Apple III Owner's Guide* and "The Storage Command" in this chapter.

## **SCROLLING**

To find your files more quickly, VisiCalc allows you to scroll through the file names in a diskette directory. For example, if you want to load a file, scroll through the local names by typing /SL and pressing →. The drive whirs and displays the first file name it finds on the edit line.

To view other file names, press → again. You can press → until you have scrolled through every file name in that directory. You can also scroll backwards by pressing ←; however, ← works only after you have pressed →. To select a file for loading, scroll to the file name you want, and press RETURN.

If you are using subdirectories, scrolling operates a little differently. After you load the program, press /SS → to put the names in the main diskette directory on the edit line. To enter a subdirectory, scroll to its name and press RETURN. You can then use → and ← to scroll through the subdirectory names. When you reach the last name in the subdirectory, press → to place the subdirectory name on the edit line. You can then type a new file name to add to the subdirectory. Typing → again will cancel the storage command.

When you enter a subdirectory, the prefix for that directory is set (see the *Apple III Owner's Manual*). The next time you scroll through file names, the names in that subdirectory appear. If you wish to enter a different directory, type a different directory prefix when you are prompted for the file name. You can also enter a different directory when a file name is on the edit line. Pressing ↑ will move the edit cue back to the previous slash, deleting the characters in between. You can press ↑ until you have deleted the subdirectory name and type in a new name, or you can press ↑ once and press → to scroll through names in the next higher directory.

Note that you can scroll file names on the edit line with /SS, /SL, /SD, S#S, S#L, and /PF.

You can also use scrolling to modify a file name already in the directory—a handy technique for updating. For example, if you load and modify a file called SALES3.VC and want the update version to be called SALES4.VC, type /SS →.

Continue to press → until the name .D1/SALES3.VC appears on the edit line. Then erase the 3 by using **ESCAPE**, replace it with 4, and reappend the .VC suffix (or use CONTROL-E to edit the file name). Press **RETURN** to save the current worksheet as .D1/SALES4.VC.

If you want to modify a pathname, you can press ↑ to jump the edit cue back to the previous slash. For example, if you have the pathname .D1/SUB1/FILE1 and you want to change SUB1 to SUB2, you would press ↑ **ESCAPE ESCAPE**, then type 2/FILE1.



Each cell on the worksheet can be assigned a local format. The label or value in a cell is not changed by the **Format** command; only their appearance is affected. The entry line shows the value or label as it was entered. The full precision of a value is used in calculations.

The entry line shows the local format. The local format is assigned to the cell contents, not the cell location; if the contents are moved to another location, the local format moves with it. Erasing the contents of a cell with the **Blank** command, however, does not remove a local format. Clearing the worksheet with the **/CA** or **/CS** options of the **Clear** command removes all local formats.

To assign a format to a row or column before writing on it, format the first entry and replicate the format with the **Replicate** command across the row or down the column.

The following example is used to illustrate all the **Format** command options. To type this example and save it for future reference, follow these steps.

Example:

Type:	Result:
1. <b>/CAY</b>	The worksheet clears, the cursor moves to A1, and the global format is set to General.
2. <b>Label entry</b> → <b>1.23456789</b> → <b>99.999 RETURN</b>	One label and two values are displayed.
3. <b>/SSformex RETURN</b>	The example is saved in a file named formex.

## **Default Format—/FD**

The Default format is the format last set with the **Global** command. If no global format is in effect, the Default is the **General** format, described later in this section.

This example (1) uses the **Global** command to display all numbers with two decimal places (**Dollars-and-Cents** format), (2) uses the **Format** command to set the local format at **C1** to **Integer**, and (3) uses the **Format** command again to default to the **Global Dollars-and-Cents** format.

Example:

<b>Type:</b>	<b>Result:</b>
1. Load or type the sample worksheet.	
2. <b>/GFS</b>	Entry Line: C1 [V] 99.999 B1: 1.23 C1: 100.00  The worksheet is displayed in <b>Dollars-and-Cents</b> format.
3. <b>/FI</b>	Entry Line: C1 /FI [V] 99.999 B1: 1.23 C1: 100  C1 is now displayed in <b>integer</b> format; B1 is unchanged.
4. <b>/FD</b>	Entry Line: C1 [V] 99.999 B1: 1.23 C1: 100.00  The <b>Default</b> format is <b>Dollars-and-Cents</b> because that was the last global format set.

Even though the entry at **C1** was displayed in three different formats, the actual value—as shown on the entry line—remained the same. The entry line value is used in calculations.

## General Format—/FG

The General format is in effect when you load the VisiCalc program or clear the worksheet. In the General format:

- Numeric values are displayed to the maximum precision and aligned with the right edge of the column. The leftmost position of each column is always blank so that values won't run together across the worksheet.
- Labels are displayed aligned with the left edge of the column. The label is displayed only to the right edge of the column, but the full label is kept in memory and displayed on the entry line.

This example uses the Global command to change the global format to Integer, and the Format command to set the local format at C1 to General.

Example:

Type:	Result:
1. Load or type the sample worksheet.	
2. /GFI	Entry Line: C1 [V] 99.999 B1: 1 C1: 100 The Integer format controls the entire worksheet and has rounded both entries.
3. /FG	Entry Line: C1 /FG [V] 99.999 B1: 1 C1: 99.999 The local format at C1 is General, so the value is displayed just as it was when the global format was General. B1, however, is still controlled by the global Integer format.

### **Integer Format—/FI**

Integer format rounds all values to the nearest whole number and displays the rounded number. Values are stored with full precision in memory and full precision values are used in all calculations.

### **Left-Aligned Format—/FL**

The Left-aligned format aligns a label with the left edge of the cell, and aligns values with the second space from the left edge of the cell. Cells which contain values reserve the left-most character space to keep values from running together across the worksheet.

This format displays values as in the General format (maximum precision); decimal values cannot be displayed both as rounded integers and left-aligned using the Format command. For information on more detailed cell displays, see the section on "The Attribute Command" earlier in this chapter.

Example:

<b>Type:</b>	<b>Result:</b>
1. Load or type the sample worksheet.	
2. <b>/FL</b>	<b>Entry Line:</b> C1 /FL (V) 99.999 <b>B1:</b> 1.234568 <b>C1:</b> 99.999  The value at C1 moves to the left side of the entry position (the first character space remains blank).
3. <b>/FI</b>	<b>Entry Line:</b> C1 /FI (V) 99.999 <b>B1:</b> 1.234568 <b>C1:</b> 100  The Integer format rounds the value at C1 to 100 and displays it aligned with the right edge of the column.
4. <b>/FL</b>	<b>Entry Line:</b> C1 /FL (V) 99.999 <b>B1:</b> 1.234568 <b>C1:</b> 99.999  The value at C1 is displayed just as it is in the General format (maximum precision) and moved to the left of cell C1 (the first character space remains blank).

**Right-Aligned Format—/FR**

The Right-aligned format displays a label or value aligned with the right edge of the cell.

Example:

<b>Type:</b>	<b>Result:</b>
1. Load or type the sample worksheet.	
2. > <b>A2 RETURN</b>	Cursor moves to <b>A2</b> .
3. <b>Label RETURN</b>	Entry Line: A2 (L) Label
4. <b>/FR</b>	Entry Line: A2 /FR (L) Label Label shifts to the right of cell <b>A2</b> .
5. > <b>A1 RETURN</b>	Entry Line: A1 (L) Label entry <b>A1:</b> Label ent
6. <b>/FR</b>	Entry Line: A1 /FR (L) Label entry <b>A1:</b> Label ent  Because the label fills all of cell <b>A1</b> , it can't move to the right. As the entry line shows, however, the Right-aligned format is assigned to the cell.
7. <b>/GC14 RETURN</b>	Now the label is moved to be aligned with the right edge of the cell because the label is shorter than the column width and can be moved within the cell.

**Dollars-and-Cents Format—/F\$**

The Dollars-and-Cents format displays all values rounded to two decimal places; it adds two decimal places (.00) to integers and an extra 0 to values with just one decimal place. A \$ is not displayed. The command has no effect on labels.

Example:

<b>Type:</b>	<b>Result:</b>
1. Load or type the sample worksheet.	
2. <b>/F\$</b>	Entry Line: C1 /F\$ (V) 99.999 <b>C1:</b> 100.00

### **Graph Format — /F\***

The Graph format displays the number of asterisks equal to the truncated (not rounded) integer value of the cell. A cell can display one fewer asterisk than the column width (the first character space is blank).

Example:

<b>Type:</b>	<b>Result:</b>
1. <b>/CAY</b>	The worksheet clears and the cursor moves to A1.
2. <b>1↓2↓3↓4↓5 &gt; B1RETURN</b>	Puts a column of numbers in column A.
3. <b>/F*</b>	Sets the Graph format at B1.
4. <b>/RRETURN↓.B5RETURN</b>	
5. <b>+ ← RETURN</b>	The formula +A1 sets B1 equal to A1.
6. <b>/R RETURN↓.B5 RETURN R</b>	Replicating the formula with a relative reference to A1 sets B2-B5 equal to A2-A5. Now the values are displayed as numbers in column A and a bar graph in column B.

Any number typed at A1-A5 is displayed as a number in column A and a bar of asterisks in column B. (A number larger than the column width displays the maximum number of asterisks, one less than the column width.) See "Drawing Bar Graphs" in Lesson Nine for more detailed examples of the graph function.

### **Repeating Labels — /F —**

You can repeat a label character across a cell by typing /F — and the character. This works the same way as the Repeating Label command (/ —). See the section below on "The Repeating Label Command" for more information.

## Assigning Complex Formats Easily — /F=

The /F= option allows you to use a letter, or shifted number to stand for a complex format that you specify with the options of the **A**tttribute command. You can specify up to 37 formats, any one of which you can invoke by typing /F and the format character.

You should follow these steps when defining your own format:

1. With the cursor in any cell, type /F=. The following prompt will appear:

```
User Format: Name: A-Z !@#$%^*()'"
```

2. Type a one-character format name. This name identifies the set of attributes you are about to enter as a format that can be invoked with /F.

3. After you enter the format name, this prompt will appear:

```
User Format: E H M T L V D O
```

This prompt lists the same options as the **A**tttribute command prompt, plus **O**. You can now select the attributes you want for this format the same way you select attributes in the /**A** command (for information on attributes, see the preceding section in this chapter, "The **A**tttribute Command").

**Note:** A format does not automatically go into effect in the cell where the format was defined. To assign the format to the cell, type /F(format name).

Whenever you type /F followed by the name of a format you defined, the attributes identified with the format name are assigned to the cell where the cursor is.

You can name formats that you define with any letter (except D), or any shifted number, quote or apostrophe (!@#\$%^\*()' '). If you name a format G, I, L, R, \$, \*, be aware that for the purposes of the worksheet where you defined it, your format takes the place of the program's standard format. If you name your format I, for instance, typing /FI will display the cell's contents according to your definition of I, not in integer format, as would usually be the case.

If you name a format G, I, L, R, \$, \*, you can go back to the original program-supplied format by typing /F=(format name)OY. You can not name a format /FD, because this is the default format. If you want to clear a format that is named something other than G, I, L, R, \$, \*, type /F=(format name)OY.

If you want to clear a cell of all formats, put the cursor in the cell and type **/FD**. This works globally, also; just type **/GFD**.

To review all the formats you have defined, type **/F**. The names of all the formats you defined on the current worksheet will appear on the edit line, under the normal Format command prompt.

### **Format to Attribute Conversions**

Formats are converted into attributes by the VisiCalc program. If you type **/F\$** in a cell, for instance, and then type **/A** to see if that cell has any attributes set in it, the Prompt Line will read:

`/A L<: V2F:`

(The section on "The Attribute Command," particularly figure 3-2, describes these settings in detail.)

## THE FUNCTIONS — @

Functions do complex calculations that would otherwise take a lot of time. Some functions save you the effort of typing frequently-used formulas (such as adding the values of a range of locations); some save you the time of using a reference table (such as trigonometric functions); some let you choose alternative values from tables (such as looking up tax rates from a table).

A function can be used anywhere a formula can be used. It consists of the @ sign, the name of the function and an argument (or list of arguments) in parentheses. An argument is a value (formula or reference to other locations) that the function uses to calculate its own value.

The @SUM function, for example, can take any number of arguments:

```
@SUM(B1,S2,A4*.23)
```

This adds the value of B1, S2, and .23 times the value of cell A4.

The @SQRT function takes one argument:

```
@SQRT(625)
```

Its value is 25, the square root of 625.

The @CHOOSE function selects one of several alternative values based on the value of the first argument:

```
@CHOOSE(A4,17,8,23,44)
```

The value of the @CHOOSE function depends on the value of its first argument, A4. If A4 is 1, the value is 17; if A4 is 2, the value is 8; and so forth.

The @ starts a value entry (no preceding + is necessary).

The functions available in the VisiCalc program are listed in this section under five headings:

- Arithmetic Functions
- Calendar Functions
- Financial Functions
- Logic Functions
- Trigonometric Functions

### **Arguments**

The following table lists the various types of arguments required by the VisiCalc functions.

<b>Argument</b>	<b>What it Means</b>
<i>v</i>	A single value; either a cell reference, a number, or an expression that evaluates to a number.
<i>list</i>	A list of single values or ranges separated by commas.
<i>range</i>	A series of adjacent cells (A1...A5 and A1...C1 are ranges).

## Arithmetic Functions

The functions in the following table perform the listed arithmetic calculation on a single argument (specified by *v* in the table).

Function	Result
@ABS( <i>v</i> )	Absolute value of the argument.
@EXP( <i>v</i> )	<i>e</i> (2.71828...) to the power specified by the argument.
@INT( <i>v</i> )	Integer portion of the argument.
@LN( <i>v</i> )	Natural log (base <i>e</i> ) of the argument.
@LOG10( <i>v</i> )	Logarithm (base 10) of the argument. ( $Y = @LOG10(X)$ ; $X = 10$ to the $Y$ power.)
@SQRT( <i>v</i> )	Square root of the argument.

### ROUNDING FOR LESS PRECISION: @ROUND (*v*, precision)

Because the VisiCalc program calculates and rounds to 11 (and sometimes 12) decimal places, differences of pennies and even dollars can occur between VisiCalc results and those produced by ordinary 2-place precision calculations. This difference can cause problems when attempting to balance books and during audits.

To solve this problem, you can use the @ROUND function to reduce the calculating precision to as many places before or after the decimal point as you want. The @ROUND function takes two arguments: the value you want rounded (*v*) and the number of decimal places you want it rounded to (the precision). The value argument can be a cell reference, a formula, or a any real number; the precision argument can be from -59 to 59.

If the precision argument is zero or a positive number, it indicates how many decimal places the number is to be rounded to. For example, @ROUND(123.456,1) equals 123.5; @ROUND(123.456,2) equals 123.46; @ROUND(123.456,0) equals 123.

If the precision argument is negative, it indicates how many places to the left of the decimal point the number is to be rounded to. For example, @ROUND(123456, -2) equals 123500. You can use the @ROUND function in each cell to round numbers on a balance sheet or other worksheet to hundreds, thousands, or millions.

You can then divide every number on the worksheet by 10, 100, 1000, and so forth, to simplify the appearance of the worksheet. You might want to make a note such as "000 omitted" at the top of such a worksheet, so you will remember the true values of the numbers.

Numbers rounded with @ROUND actually change in memory; the rounded values are used in all calculations. Numbers are rounded to the specified precision, but trailing zeroes are not displayed. (@ROUND(109.84499,4) equals 109.845, for example.) This is unlike cells with value display attributes, which affect only how numbers are displayed.

#### **THE MODULO FUNCTION — @MOD(*n,m*)**

The Modulo function takes two arguments; the function returns the remainder after the first argument (*n*) is divided by the second argument (*m*). Both arguments must be real numbers, but need not be integers. The second argument (*m*) may not be 0. The value returned has the same sign (+ or -) as the second argument (*m*).

#### **THE DOT PRODUCT FUNCTION — @DOTPROD(*range1,range2*)**

The Dot Product function multiplies the contents of corresponding cells in two ranges and sums the products. For instance, @DOTPROD(A1...A5,B1...B5) yields a value equal to (A1\*B1) + (A2\*B2)... + (A5\*B5). Ranges may be from either columns or rows; you can mix column and row ranges. There must be an equal number of items in each range, and labels in any range evaluate to 0.

**ARITHMETIC FUNCTIONS THAT USE A LIST OF ARGUMENTS**

The functions in the following table do calculations with a list of arguments (represented by "list" in the table). The list can have numerous elements. The arguments are separated with commas.

<b>Function</b>	<b>Result</b>
@AVERAGE(list)	Arithmetic mean of the values in the list. The result is equivalent to @SUM(list) divided by @COUNT(list). The maximum number of elements in the list is 255.
@COUNT(list)	Number of non-blank entries in the list. The maximum number of elements in the list is 255.
@MAX(list)	Largest value in the list.
@MIN(list)	Smallest value in the list.
@SUM(list)	Sum of all values in the list.

## **ARITHMETIC FUNCTIONS WITHOUT ARGUMENTS**

Several arithmetic functions do not require an argument.

**@PI:** @PI is the ratio of the circumference of a circle to its diameter, 3.1415926536.

**@NA and @ERROR:** @NA (Not Available) is used when a worksheet is set up before the data is written. Because a blank cell evaluates to 0, ERROR is displayed in each cell where zero appears as a denominator and incorrect or misleading values can be produced at other cells. Writing @NA in the blank cells causes the VisiCalc program to evaluate all entries that refer to those positions as NA.

The result of an illegal calculation is displayed on the worksheet as ERROR. This can be caused several ways: (1) by a too-deep nesting of ( or + , such as +A1\*((B2-(B3+(B4\*(B5-(B3/(B1+(B1/(D1-(B2+(B7\*(B9+B10)/(B4\*(B7+(B1-B2))))))))))))); (2) by an error in writing a formula, or (3) by deleting a row or column that is referenced in a formula. ERROR is displayed in the cell that contains the error and in all locations that refer to it.

The @ERROR function causes ERROR to be displayed at the location where it is entered and all locations that refer to it.

**@TRUE and @FALSE:** @TRUE and @FALSE are used with the logic functions described later in this section. They cause TRUE or FALSE to be displayed at the locations where they are entered. The values TRUE and FALSE are also displayed when the comparison operators (<, >, =, <=, >=, and <>) are used.

## **FUNCTIONS THAT SELECT ALTERNATIVE VALUES AND LABELS**

Three functions can be used to select alternatives for calculations.

**@CHOOSE:** The @CHOOSE function takes one of the values in its list of arguments. The first element in the list is the index to the following arguments. For example, in @CHOOSE(A4,17,6,33,39), A4 is evaluated first. If A4 is 1, the result is 17; if A4 is 2, the result is 6; and so on. @CHOOSE evaluates to NA if the first argument is zero or less, or if the value of the first argument is greater than the number of remaining arguments.

**@LCHOOSE:** The @LCHOOSE function works just like the @CHOOSE function, except that @LCHOOSE can return a label instead of a value. @LCHOOSE takes one of the values in its list of arguments. The first argument in the list is the index to the following arguments. For example, in @LCHOOSE(A1,B4,B5,B6), A1 is evaluated first. If A1 is 1, the result is the contents of cell B4; if A1 is 2, the result is the contents of cell B5; and so on. @LCHOOSE evaluates to NA if the first argument is zero or less, or if the value of the first argument is greater than the number of remaining arguments.

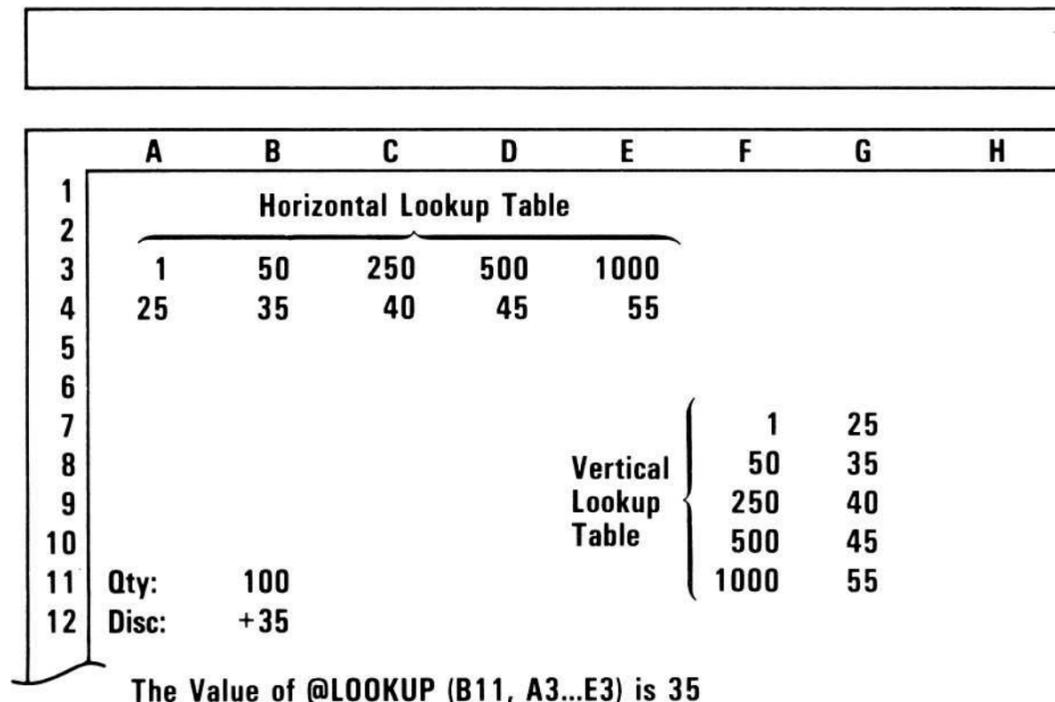
Note that @LCHOOSE *always* returns the contents of a cell, whether the cell is a label or a value. "@LCHOOSE(expression)" is the same as "@LABEL(@CHOOSE(expression))" (see the following section on @LABEL for details).

**@LOOKUP:** @LOOKUP searches for a value in a range of locations and evaluates to a corresponding value from an adjacent range. The range to be searched can be in either a row or a column; the adjacent range must be in the column to the right of the range searched or in the row below the range to be searched. Values in the range must be in ascending order.

The search value is compared to successive values in the search range until the largest value is found that is equal to or less than the search value (or until the end of the table is reached). The corresponding entry in the adjacent range is the value that the @LOOKUP function assumes. If the value being searched for is smaller than the first entry in the lookup table, @NA is returned. If the value being searched for is larger than any value in the lookup table, the last value in the table is returned.

Two arguments are required: the first is the value to be searched for, and the second specifies the range that contains the table of values to be searched.

Figure 3-10 shows several lookup tables.



010-008

**Figure 3-10. Lookup Tables**

**THE @LABEL AND @VALUE FUNCTIONS**

**@LABEL:** @LABEL returns the expression it takes as its argument in the form of a label. The @LABEL function takes an expression as its argument. The expression can be simple (A1) or complex (@IF(A1 > A3, @LOOKUP(B1, B2...B5), D5)). Since @LABEL is a function, and therefore a value entry, however, a cell that contains an @LABEL function is still a value cell.

This type of "hybrid entry" lets you display the result of a function, such as @IF, in the form of a label. For instance, consider the formula @IF(1 = 2, A1, A2), which returns A2 since 1 = 2 is FALSE. If the label cat is in cell A2, the value of the formula would be 0, because normally a cell with a label in it has the value 0. However, if the formula was @LABEL(@IF(1 = 2, A1, A2)), the formula would return @LABEL(A2). In this example, the cell with the formula in it would display cat. If A2 contained the mathematical expression "@SUM(B1...B5)," @LABEL(A2) would return the sum of the values in B1 through B5.

When you use this function with @LOOKUP, you can achieve the effect of having @LOOKUP return a "hybrid entry" that looks like a label but is a value. Thus, in a lookup table that returns discount rates based on the amount of goods purchased, you could make the @LOOKUP function return "Five percent."

@LABEL has several subtle qualities. First, a cell with an @LABEL(*coordinate*) function in it looks like a label cell if the cell referenced by *coordinate* contains a label. Since the cell with the @LABEL in it is a value cell, however, the cell is counted by functions such as @COUNT, @MAX, @AVERAGE, and @MIN. The @COUNT function, for example, does not count cells with labels in them; a cell containing @LABEL(A1) would be counted, however, even if A1 contained a label. Second, @LABEL does not change a value into a label;  $1 + @LABEL(A1)$  is still 2 if A1 contains the value 1. This is also true for the special values: @NA, @ERROR, @TRUE and @FALSE.

**@VALUE:** The @VALUE function guarantees that a cell reference will return a "true value," and not a "hybrid entry" (see above). As mentioned above, @LABEL returns a "hybrid entry" that looks like a label but is a value; @VALUE allows you to insure that a "hybrid entry" does not creep into a formula. For instance, consider the formula given above: @LABEL(@IF(1=2,A1,A2)). This formula always returns a "hybrid entry." If you change the formula to @LABEL(@IF(1=2,A1,@VALUE(A2))), however, a true value, as opposed to a hybrid entry, is returned. Note that even inside an @LABEL function, @VALUE guarantees that a true value is returned.

## Calendar Functions

Calendar functions perform date arithmetic. You can use the calendar functions for simple applications, such as determining the date 20, 50, 90, or 100 days from today. You can also use the calendar functions for more advanced applications, especially in conjunction with the financial functions. An example of an advanced application is determining the accrued interest on a bond, certificate of deposit, or commercial paper instrument.

**CONVERTING CALENDAR DATES TO ABSOLUTE DATES—@VMDY AND @MDY**

You can convert calendar dates to *absolute dates* by using the @VMDY or @MDY functions. An absolute date is the number of days between the date you give and a fixed date. In the VisiCalc program, the fixed (base) date is Monday, January 1, 1979. Thus, @VMDY(1,1,1979) is 0; @VMDY(1,2,1979) is 1; @VMDY(1,3,1979) is 2; @VMDY(1,1,1980) is 365, and so on.

Both @VMDY and @MDY take leap years into consideration when doing calendar arithmetic.

**The Month-Day-Year Function—@MDY:** @MDY returns the number of days since the base date of January 1, 1979. @MDY takes three arguments, which must be separated by commas (,). The arguments are presented in the following list, in the order you enter them in the function:

- **Month.** Must be an integer. Single-digit months do not have to be preceded by a zero. Negative integers refer to months in the previous year; for example, @MDY(-1,1,1982) returns the same number as @MDY(11,1,1981)—1035. Likewise, @MDY(0,1,1982) returns the same number as @MDY(12,1,1981).
- **Day.** Can be any real number. Single-digit days do not have to be preceded by a zero. Decimal numbers refer to fractions of a day. Negative days refer to days in the previous month; for example, @MDY(2,-1,1982) returns the same number as @MDY(1,30,1982). Likewise, @MDY(2,0,1982) returns the same number as @MDY(1,31,1982). Also, dates that go past the last day in the month are carried over to the next month. @MDY(2,31,1982), for instance, is the same as @MDY(3,3,1982).
- **Year.** Must be a positive integer. You cannot abbreviate the year; it must be typed 1981 or 1985, not 81 or 85. @MDY(1,1,79) returns the absolute date of January 1 in the year 79 A.D., while @MDY(1,1,1979) returns the absolute date of January 1, 1979.

@MDY has many interesting uses. It can be used to determine the number of days between two dates, or the day of the week of a particular date (when used with @MOD). For example, (@MDY(6,6,1945) – @MDY(12,7,1941)) returns the number of days in World War II. @MOD(@MDY(7,4,1776),7) returns the day of the week the Declaration of Independence was signed. (0 is Monday, 1 is Tuesday, etc.).

@MDY can also be used in conjunction with @DAY, to obtain the calendar date of the last day in any month. The form is: @DAY(@MDY(N + 1,0,year), where N is the number of the month whose last calendar day you're trying to find. @DAY(@MDY(3,0,1984)), for instance, returns 29, which means that the last day in month 2 of 1984 is the 29th day in the month. (February is a leap year in 1984.) What you're really asking for is the 0th day in month 3 (March), which @MDY converts to the last day in month 2 (February). Thus 29 is returned.

There are many other ways to use the calendar functions together; this is merely one application.

**The Valid Month-Day-Year Function—@VMDY** @VMDY works the same as @MDY, except it takes only real dates as arguments. For example, @VMDY(-1,0,1979) returns ERROR. This is in contrast to @MDY, which is described in the previous section.

@VMDY takes three arguments:

- Month. Must be an integer, from 1 to 12. Single-digit months do not have to be preceded by a zero.
- Day. Can be any real number, from 1 to 31. Single-digit days do not have to be preceded by a zero. Decimal numbers refer to fractions of a day.
- Year. Must be a positive integer. The year cannot be abbreviated; you must type 1984 or 1981, not 84 or 81. @VMDY(1,1,79) is January 1 in the year 79 A.D., while @VMDY(1,1,1979) is January 1, 1979.

### THE DAY FUNCTION—@DAY

@DAY takes an absolute date as its argument. It returns the day portion of the calendar date corresponding to the absolute date given as the argument. For instance, @DAY(100) is 11, because 4,11,1979 is 100 days after January 1, 1979 (the base date used for all calendar function calculations).

### **THE MONTH FUNCTION—@MONTH**

@MONTH takes an absolute date as its argument. It returns the month portion of the calendar date corresponding to the absolute date given as the argument. For instance, @MONTH(100) is 4, because 4,11,1979 was 100 days after January 1, 1979 (the base date used for all calendar function calculations).

### **THE YEAR FUNCTION—@YEAR**

@YEAR takes an absolute date as its argument. It returns the year portion of the calendar date corresponding to the absolute date given as the argument. For instance, @YEAR(370) is 1980, because 1,6,1980 was 370 days after January 1, 1979 (the base date used for all calendar function calculations).

### **THE HOURS, MINUTES, SECONDS FUNCTION—@HMS**

The time functions treat time as a fraction of a 24-hour day, counted from midnight. Thus, .25 represents 6:00 AM.

@HMS converts a time argument (hours, minutes, seconds) into fractions of a day. @HMS(12,0,0) returns .5, for example, since 12 hours, 0 minutes and 0 seconds is one-half of a day.

Hours are entered on the basis of a 24 hour day; 1 PM, for instance, is entered as 13. @HMS takes hour arguments larger than 24, converting into the appropriate number of days. For example, @HMS(36,0,0) returns 1.5, because 36 hours equals one-and-one-half days. @HMS also takes minute and second arguments larger than 60.

Arguments do not have to be integers. For example, @HMS(12.5,0,0) is the same as @HMS(12,30,0). The third argument, seconds, is optional. If no argument for seconds is given, 0 is assumed.

@HMS also takes negative numbers as arguments and converts them to valid times. For example, @HMS(23,60,-60) is the same as @HMS(23,59,0)—both are equal to .993056, or about 99/100ths of a day.

@HMS can also be used with the trigonometric functions to convert Hours, Minutes, Seconds angle notation to radians.

**THE HOUR FUNCTION—@HOUR**

@HOUR takes an absolute date and returns the hour portion of the corresponding calendar date. For instance, @HOUR(100.5) returns 12, because one-hundred and one-half days after January 1, 1979 (the base date for all absolute dates), it was 12 o'clock.

**THE MINUTE FUNCTION—@MINUTE**

@MINUTE takes an absolute date and returns the minute portion of the corresponding calendar date. For instance, @MINUTE(100.52083) returns 30, because 100.52083 days after midnight, January 1, 1979, (the base date for all absolute dates), it was 12:30.

**THE SECOND FUNCTION—@SECOND**

@SECOND takes an absolute date and returns the second portion of the corresponding calendar date. For instance, @SECOND(100.52135) returns 45, because 100.52135 days after midnight, January 1, 1979, (the base date for all absolute dates), it was 12:30:45 (three-fourths of a minute past 12:30).

**Financial Functions**

Financial functions perform financial calculations, such as Present Value, Future Value, and Internal Rate of Return.

The following sections are necessarily brief; they do not teach the principles behind financial calculations. If you want to learn more about financial calculations, you might wish to consult a Finance or Accounting textbook, for example:

- Weston, J. Frederick, and Brigham, Eugene, **MANAGERIAL FINANCE** (Hinsdale, Illinois: The Dryden Press, 1981, Sixth Edition).
- Viscione, Jerry A., **FINANCIAL ANALYSIS** (Boston, Massachusetts: Houghton Mifflin, 1977).
- Davidson, Sidney; Stickney, Clyde P.; and Weil, Roman, **FINANCIAL ACCOUNTING** (Hinsdale, Illinois: The Dryden Press, 1981, Second Edition).

Financial functions take four arguments. Abbreviations for arguments to the Financial functions are presented in the following table.

n	Number of periods; must be an integer greater than zero.
pmt	Payment each period; can be positive or negative.
i	Interest rate; must be in decimal form (e.g. 10% = .10); may be positive or negative but not zero.
pv	Present value of a series of payments. May be positive or negative but not a range.
fv	Future value of a series of payments. May be positive or negative but not a range.

The Financial functions are presented in the table below (arguments are separated by commas).

Function	Result
@RATE(n,pmt,pv,fv)	Interest or discount rate of the given arguments.
@PERIODS(i,pmt,pv,fv)	Number of periods of the given arguments.
@PMT(i,n,pv,fv)	Amount of each payment of the given arguments.
@PV(i,n,pmt,fv)	Present Value of an <i>even</i> series of payments.
@FV(i,n,pmt,pv)	Future Value of an <i>even</i> series of payments.
@IRR( <i>investment,range</i> )	Internal Rate of Return based on an initial <i>investment</i> and a <i>range</i> of even or uneven cash flows. Both arguments must be supplied.
@NPV(i, <i>range</i> )	Net Present Value of a series of even or uneven cash flows (cfl...cfn); this series is the <i>range</i> argument.

The formulas used in the Financial functions are listed in Appendix D.

For the purposes of Financial functions, amounts (payments or investments) have a negative value and all amounts received (loans or investment income) have a positive value. Negative interest rates indicate deflation.

When a financial function takes periods ( $n$ ) and/or interest rate ( $i$ ) as arguments, they must be specified. This is true for @RATE, @PERIODS, @PMT, @PV, @FV, and @NPV. The period ( $n$ ) must be given as an integer value. For information on calculating payment (@PMT) or future value (@FV) with a fractional period, see Appendix D.

Arguments must be values, formulas, or cell references. Of the arguments other than  $i$  and  $n$ , one can be @NA; all four may be entered if they are known. Two exceptions to this are the Internal Rate of Return Function and the Net Present Value function; both require only two arguments which *must* be supplied.

The Financial functions calculate all values based on the *end* of the period. This is important for the @PMT, @PV, and @FV functions. To convert a result from payments made at the end of the period to payments made at the beginning of the period, multiply the result by  $(1 + i)$ .

### INTEREST OR DISCOUNT RATE — @RATE

The Interest or Discount Rate function calculates the interest rate (for loans) or discount rate (for annuities) based on the following arguments, in order: number of periods, payment each period, present value of the payments, future value of the payments.

The periods argument ( $n$ ) must be supplied. One of the following arguments may be entered as @NA if all other arguments are specified: payment (pmt), future value (fv), present value (pv).

Acceptable combinations of arguments are:

@RATE( $n$ , pmt, pv, @NA)

@RATE( $n$ , pmt, @NA, fv)

@RATE( $n$ , @NA, pv, fv)

@RATE( $n$ , pmt, pv, fv)

### **NUMBER OF PERIODS – @PERIOD**

The Number of Periods function calculates  $n$ , the number of periods required to satisfy the arguments given to the function. Its arguments are, in order: interest rate in decimal form, payment, present value, future value. This function returns a real number.

The interest rate argument ( $i$ ) must be specified. One of the following arguments may be entered as @NA if all other arguments are specified: payment ( $pmt$ ), future value ( $fv$ ), present value ( $pv$ ).

Acceptable combinations of arguments are:

@PERIODS( $i, pmt, pv, @NA$ )  
@PERIODS( $i, pmt, @NA, fv$ )  
@PERIODS( $i, @NA, pv, fv$ )  
@PERIODS( $i, pmt, pv, fv$ )

### **PAYMENT – @PMT**

The Payment function calculates the payment that must be made each period to satisfy the requirements of the arguments given to the function. These arguments are, in order: interest rate in decimal form, number of periods, present value, future value.

The period ( $n$ ) and interest rate ( $i$ ) arguments must be specified. The period ( $n$ ) must be given as an integer value. For information on calculating payment with a fractional period, see Appendix D. One of the following arguments may be entered as @NA if the other argument is specified: future value ( $fv$ ), present value ( $pv$ ).

Acceptable combinations of arguments are:

@PMT( $i, n, pv, @NA$ )  
@PMT( $i, n, @NA, fv$ )  
@PMT( $i, n, pv, fv$ )

**FUTURE VALUE—@FV**

The Future Value function calculates the future value of a series of payments. Its arguments are, in order: interest rate in decimal form, number of periods, payment, present value.

The period (n) and interest rate (i) arguments must be supplied. The period (n) must be given as an integer value. For information on calculating future value with a fractional period, see Appendix D. One of the following arguments may be entered as @NA if the other argument is specified: payment (pmt), present value (pv).

Acceptable combinations of arguments are:

@FV(i,n,pmt,@NA)  
@FV(i,n,@NA,pv)  
@FV(i,n,pmt,pv)

**PRESENT VALUE—@PV**

The Present Value function calculates the present value of a series of level payments. The present value of a series of payments is a single amount of money at the present time that is the economic equivalent of the entire series.

The Present Value function takes as its arguments, in order: interest rate in decimal form, number of periods, payment, future value.

The period (n) and interest rate (i) arguments must be supplied. One of the following arguments may be entered as @NA if the other argument is specified: payment (pmt), future value (fv).

Acceptable combinations of arguments are:

@PV(i,n,pmt,@NA)  
@PV(i,n,@NA,fv)  
@PV(i,n,pmt,fv)

**INTERNAL RATE OF RETURN — @IRR**

The Internal Rate of Return function returns an interest rate at which a discounted series of payments equals an initial investment. This function is a special case of the @NPV function. The @IRR function returns an interest rate for which the net present value of the discounted payments is equal to the initial investment. Since the initial investment is usually a negative cash flow, it is entered as a negative value.

This function takes two arguments, in order: investment (beginning value), and *range*. The investment argument can be either a cell or a value; it represents the initial investment for which you are calculating the rate of return. The *range* argument is a range of cells that contain the payments.

@IRR is the only financial function that can have multiple solutions. @IRR returns only the first rate at which NPV is 0, although there may be more than one. Appendix D, "Formulas Used to Calculate the Financial Functions," lists the formulas used to calculate the financial functions. If you get a different result from a financial calculator than you do with @IRR, the formulas being used may be different.

In some cases @IRR returns a value of @ERROR. Check to see that the arguments are all correctly given. If the function still returns @ERROR, it means that the VisiCalc program cannot calculate this particular situation with the @IRR function. You need to use another method to get the answer.

One alternative is to use the formula:

initial investment + @NPV(rate, cfl...cfn)

Specify the initial investment (generally a negative value) and the range of cash flows (cfl...cfn). Enter the rate as a cell reference. Then, change the rate in that cell until the formula comes close to or equals zero. That rate is the internal rate of return.

## **NET PRESENT VALUE – @NPV**

The @NPV function calculates the net present value of future cash flows. It takes two arguments, in order: the discount rate (or cost of money used to discount the future cash flows), and a range of locations that includes the cash flows themselves.

## **Logic Functions**

A logical value is one whose value is either TRUE or FALSE. A logic function is one that manipulates logical values. Logical calculations are similar to mathematical operations, but operate only on this special set of values.

**Comparison Operators:** The comparison operators work on two numeric values and evaluate to a logical value. The equal (=) and not equal (<>) operators can be used to compare logical values. For example, the formula 4 > 1 (four is greater than one) evaluates to the logical value TRUE. The formula 5 = 3 (five is equal to three) evaluates to the logical value FALSE.

These comparisons can be used either as entries on the worksheet or arguments in a logic function. If used as entries on the worksheet, they should not be written at a location referenced by a function that does not use logic functions. If this happens, the value of such functions is ERROR.

The following table lists the comparison operators (it assumes the operator is preceded by *value1* and is followed by *value2*).

Operator	Value
<	TRUE if <i>value1</i> is less than <i>value2</i> ; FALSE if it is not.
>	TRUE if <i>value1</i> is greater than <i>value2</i> ; FALSE if it is not.
=	TRUE if <i>value1</i> is equal to <i>value2</i> ; FALSE if it is not.
< =	TRUE if <i>value1</i> is less than or equal to <i>value2</i> ; FALSE if it is not.
> =	TRUE if <i>value1</i> is greater than or equal to <i>value2</i> ; FALSE if it is not.
< >	TRUE if <i>value1</i> is not equal to <i>value2</i> ; FALSE if it is.

**@NOT:** @NOT takes a single logical value as its argument (i.e., one whose value is TRUE or FALSE); the function's value is the opposite logical value. The value of @NOT(A1) is FALSE if A1 is TRUE and TRUE if A1 is FALSE. If the value of A1 is @NA, the value of @NOT(A1) is also @NA. If the value of the argument is anything other than TRUE, FALSE, or @NA, the value of the @NOT function is ERROR.

**@AND:** @AND takes any number of arguments, each of which must be a logical value or a range of logical values. Its value is TRUE if all the arguments are TRUE, FALSE if any of the arguments is FALSE. If any of the arguments is not logical or evaluates to ERROR, the value of @AND is ERROR. If any of the arguments evaluates to NA and all other arguments are TRUE or FALSE, the value of @AND is NA.

**@OR:** @OR takes any number of arguments, each of which must be a logical value or a range of logical values. Its value is TRUE if any of the values is TRUE, and FALSE if all the values are FALSE. If any of the arguments is not logical or evaluates to ERROR, the value of @OR is ERROR. If any of the arguments evaluates to @NA and all other values are TRUE or FALSE, the value of @OR is NA.

**@IF:** @IF takes three arguments. The first must be a logical value; the second and third can be any value. The function evaluates to the value of the second or third argument, depending on the value of the first.

<b>Value of First Argument</b>	<b>Evaluates to</b>
@TRUE	Value of second argument
@FALSE	Value of third argument
@NA	@NA
Not logical or @ERROR	@ERROR

For example, the value of @IF(D5,2,3) is 2 if D5 is TRUE and 3 if D5 is FALSE. The value of @IF(D5,E1,E2) is the value of E1 if D5 is TRUE and the value of E2 if D5 is false.

**@ISNA and @ISERROR:** @ISNA takes one argument. Its value is TRUE if the value of the argument is @NA, and FALSE if the value of the argument is anything else.

@ISERROR takes one argument. Its value is TRUE if the value of the argument is ERROR, and FALSE if the value of the argument is anything else.

@ISNA and @ISERROR are used to manipulate entries written as @NA or @ERROR and return a value that is not automatically designated NA or ERROR.

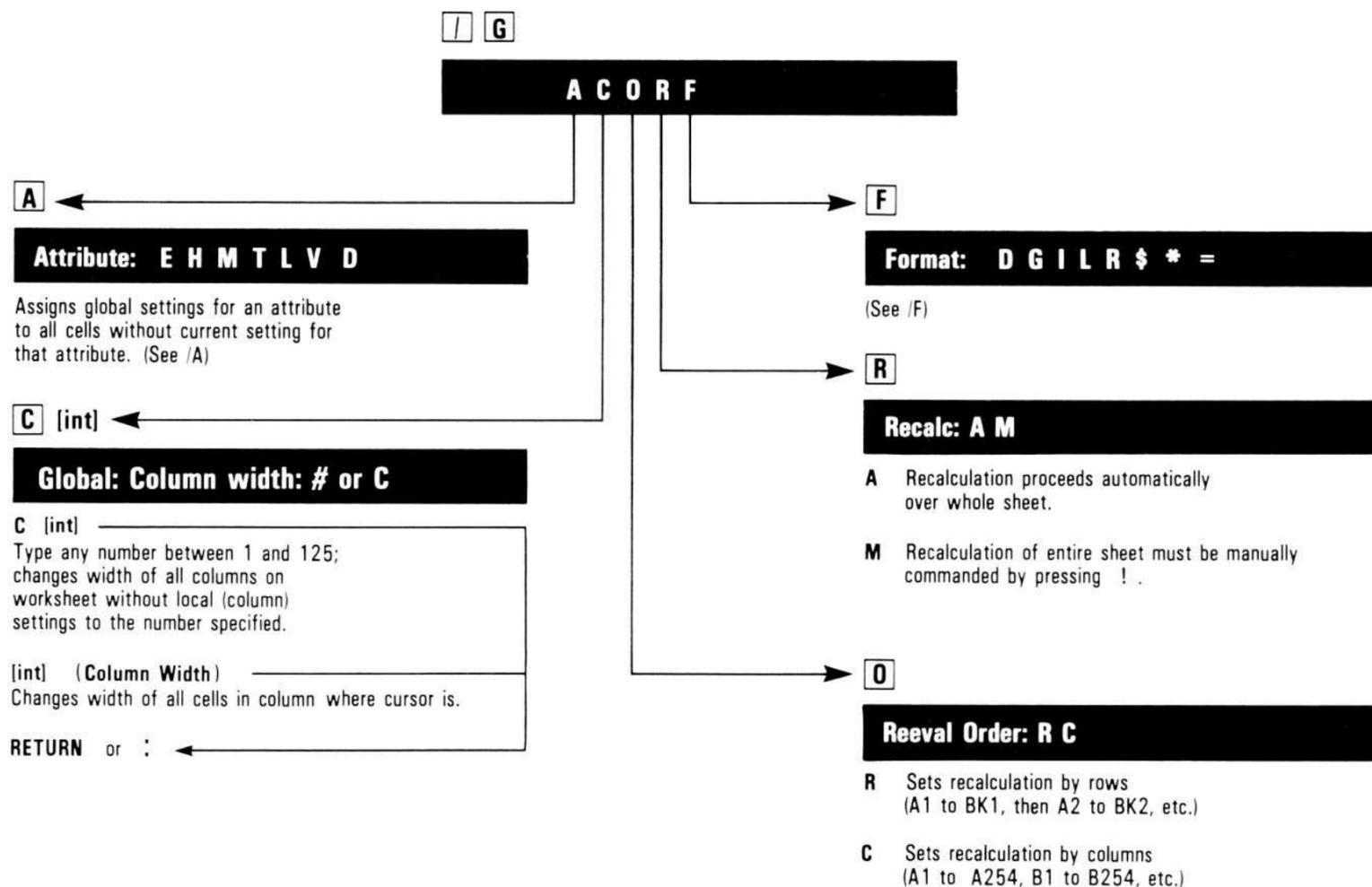
## **Trigonometric Functions**

The functions in the following table perform the listed trigonometric calculation on a single argument. All angles are specified in radians (2 pi radians = 360 degrees).

<b>Function</b>	<b>Result</b>
@SIN(v)	Sine of the argument.
@COS(v)	Cosine of the argument.
@TAN(v)	Tangent of the argument.
@ASIN(v)	Arc sine of the argument.
@ACOS(v)	Arc cosine of the argument.
@ATAN(v)	Arc tangent of the argument.

Trig functions can also take arguments in hours, minutes, and seconds with the @HMS function.

## GLOBAL COMMAND—/G



The Global command affects the entire display, unless the screen is split into two windows, in which case it affects the window that contains the cursor. The command has five options:

- A Attribute
- C Column width
- O Order of recalculation
- R Recalculation priority (Automatic or Manual)
- F Format

## **Attributes — /GA**

The Attribute option assigns the specified attributes to each cell on the worksheet that has no local attributes. The Global attributes that can be assigned are the same as the local attributes in the section of this chapter on "The Attribute Command."

If the screen is split into two windows, each window can have its own set of global attributes.

The startup, local, and global settings are interrelated. Their relationship is shown in Figure 3-3 in the "Attribute Command" section of this chapter.

## **Column Width — /GC**

The Column Width option changes the width of all columns in a window. The minimum width is 1; the maximum width is 125. Setting the column width to half the maximum or more removes vertical titles.

Individual column widths are set with /GCC, which is described in the "Single Column Width" section below.

As many columns as possible are displayed.

Changing the column width changes the way some entries are displayed. Labels longer than the column width are truncated. Values can be affected in one of three ways:

- Digits to the right of the decimal are rounded as necessary to fit in the cell.
- Numbers are displayed in scientific notation if this allows greater precision.
- A cell is filled with > signs if the number is too large to be represented.

These changes affect only the way entries are displayed. Values and labels are unchanged in memory. The entry line displays the full value or label at the cursor location, regardless of how the entry is displayed.

To put the value appearing on the entry line onto the edit line, type #. The prompt line must be clear to do this.

Example:

Type:	Result:
1. /CAY	The worksheet clears, the cursor moves to A1, and the column width is set to 9.
2. This line is too long. RETURN	Entry Line: A1 (L) This line is too long. A1: This line
3. /G	Prompt Line: Global: A C O R F
4. C	Prompt Line: Global: Column width: # or C
5. 18 RETURN	Entry Line: A1 (L) This line is too long. A1: This line is too l
6. . /GC22 RETURN	The entire label is displayed.

Example:

Type:	Result:
1. /CAY	The worksheet clears and the cursor moves to A1. Column width is 9.
2. 123456789 RETURN	Entry Line: A1 (V) 123456789 A1: 1.2346E8  The first character position is blank. The number is displayed in scientific notation, with the final digit rounded.
3. → 1.23456789 RETURN	Entry Line: B1 (V) 1.23456789 B1: 1.234568
4. /GC3 RETURN	Entry Line: B1 (V) 1.2345678 A1: >> B1: 1  A1 cannot display the number in any form. The number at B1 is rounded.

## Single Column Width—/GCC

The Single Column Width option sets the width of all cells in a column. Column widths may range from 1 to 125 characters; every column on a worksheet may have a different width. If no local column width is specified in a cell, the width will be the global default width. If no global default width has been specified, the General default value of 9 will be in effect.

To reset an individual column width to the current global default, type /GCCD.

The following example demonstrates the use of columns with differing widths.

Example:

Type:	Result:
1. /CAY	The worksheet clears, the cursor moves to A1, and the column width is set to 9.
2. /GCC15 RETURN	Column width for column A changes to 15.
3. This is a Label RETURN	Entry Line: A1 [L] This is a Label
4. →/GCC1 RETURN	Column width for column B changes to 1.
5. 3RETURN	Entry Line: B1 [V] 3

Columns that are less than 3 characters wide have these unusual display characteristics:

- Any gutters that have been set will be ignored.
- For column widths of 1, column headings beginning with AB, AC, and so on, will display only the latter letter, e.g., B will be displayed when you are in column AB. This can be confusing if you are using vertically split windows.

## Order of Recalculation Option — /GO

The Order of Recalculation option lets you recalculate the worksheet by rows or by columns. The order of recalculation is noted by the letter in the upper-right corner of the screen—C for Column, R for Row. When the VisiCalc program is loaded, or when a worksheet is cleared, the recalculation order is set to columns.

When the VisiCalc program calculates in *column* order, it begins at A1 and works down column A until it comes to the final entry in that column. Then, it moves to B1 and works its way down to the last entry in that column. It continues to calculate each column until all values are calculated.

When the VisiCalc program calculates in *row* order, it begins at A1 and calculates across row 1 until it comes to the final entry in that row. Then it moves to A2 and works its way across to the last entry in that row. It continues to calculate each row until all values are calculated.

If the calculation at any cell is incorrect, you might have a formula that is calculated before the values it refers to are calculated (a condition called a *forward reference*). Order of calculation in formulas is described under "Values" in this chapter. "Forward and Circular References" in Lesson One describes these conditions and how to avoid them.

**Example:** The following example requires column-order recalculation to produce correct results, and shows what happens when the worksheet is recalculated by rows.

Type:	Result:
1. /CAY	The worksheet clears, the cursor moves to A1, and column-order recalculation is set.
2. 1 ↓ +A1 RETURN > B1 RETURN +A2 ↓ +B1 RETURN	

3. **> A1 RETURN 2 RETURN 3 RETURN** A1 changes to 2 and 3, respectively, and the other values are recalculated. The screen should look like Figure 3-11.



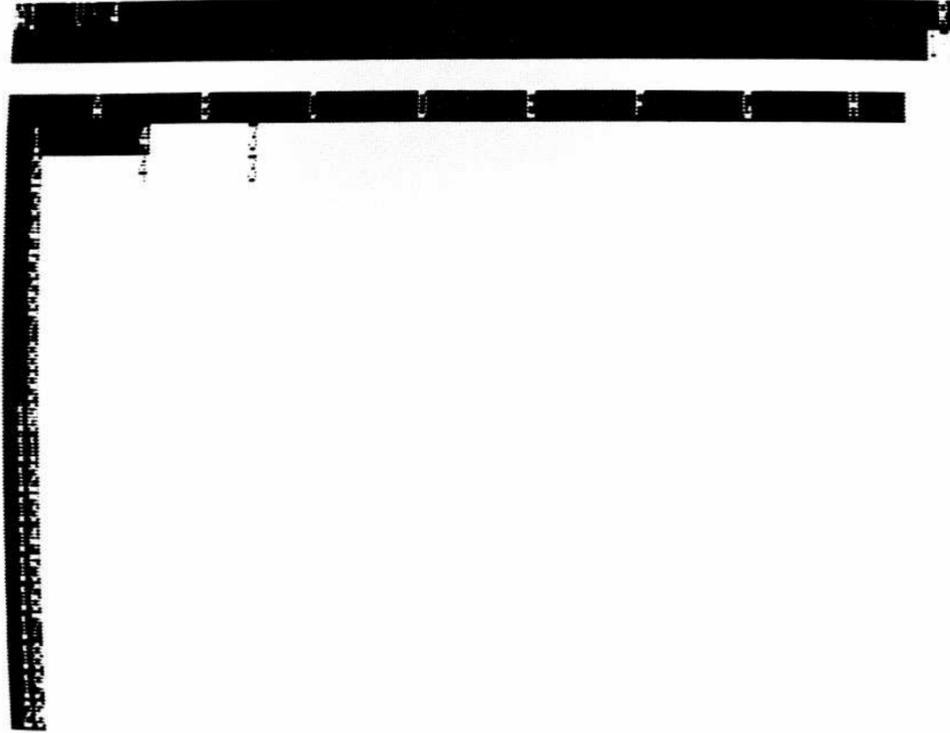
010-028/P

**Figure 3-11. Recalculation by Column (Correct Result)**

4. **/GOR** The order of recalculation indicator changes to R.

## 5. 4 RETURN

The screen should look like Figure 3-12.



010-029/P

**Figure 3-12. Recalculation by Rows (Incorrect Result)**

6. !                      The forced recalculation corrects all cells.

The values at B1 and B2 are incorrect. B1 and B2 should be the same as A1 and A2, but they display a different value because B1 was calculated before A2 was calculated. To get a correct result, you must force a recalculation by typing ! when the prompt line is blank.

The ERROR value doesn't necessarily display when you create a forward or circular reference. If you save the worksheet and reload it, ERROR does display at each location that contains a forward or circular reference. In this case, you must also force a recalculation by typing ! as many times as necessary until values do not change on the worksheet.

**NOTE:** Because forward and circular references can make it difficult to use a worksheet, you should avoid writing them, and correct them when you find them. Also, be aware that when you change the order of recalculation you can create forward references. If you load a worksheet and the ERROR value is displayed in any cell, check the formula in that cell for forward or circular references unless you know the ERROR value is caused by division by zero.

If your worksheet requires row-order recalculation to produce correct results, a column-order recalculation will produce incorrect results.

### **Recalculation Priority Option — /GR**

The Recalculation Priority option specifies whether the VisiCalc program recalculates the worksheet each time a value is typed, or whether the worksheet is recalculated only when ! is typed. Manual recalculation speeds entry of columns or rows of figures by eliminating the automatic recalculation after each entry.

In response to /GR, the prompt line reads Global: Recalc: A M The characters following Recalc: are options:

A Automatic  
M Manual

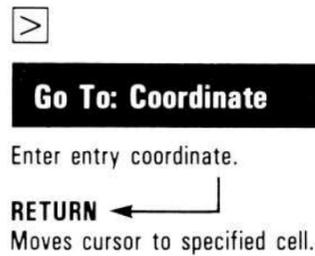
To select manual recalculation, type /GRM. To restore automatic recalculation, type /GRA.

### **Format Option — /GF**

The Format option assigns a format to every cell on the worksheet that does not have a local format. The global formats that can be assigned are the same as those for local formats; see "Format Command" in this chapter for their description. A brief summary of their keystrokes is: /GFG, /GFI, /GFL, /GFR, /GF\$, /GF-, /GF\*, /GF=, /GFO, and /GFD.

If the screen is split into two windows, each can have a different global format.

## GO TO COMMAND— >



The Go To command moves the cursor directly to a cell. It is invoked by typing >; the prompt line asks for the coordinate of the destination.

Example:

Type:	Result:
1. >	Prompt Line: Go to: Coordinate
2. B20 RETURN	The cursor moves to location B20 (column B, row 20).

## THE HELP FACILITY—?

You can get information about all commands in the VisiCalc program at any time by typing ?. The information displayed after you type ? describes your options from the point in the program where you asked for help. For example, if you are in the middle of replicating when you ask for help, the Help text displayed describes the replicate command. You can also use the Help Facility to find out where you are in the program.

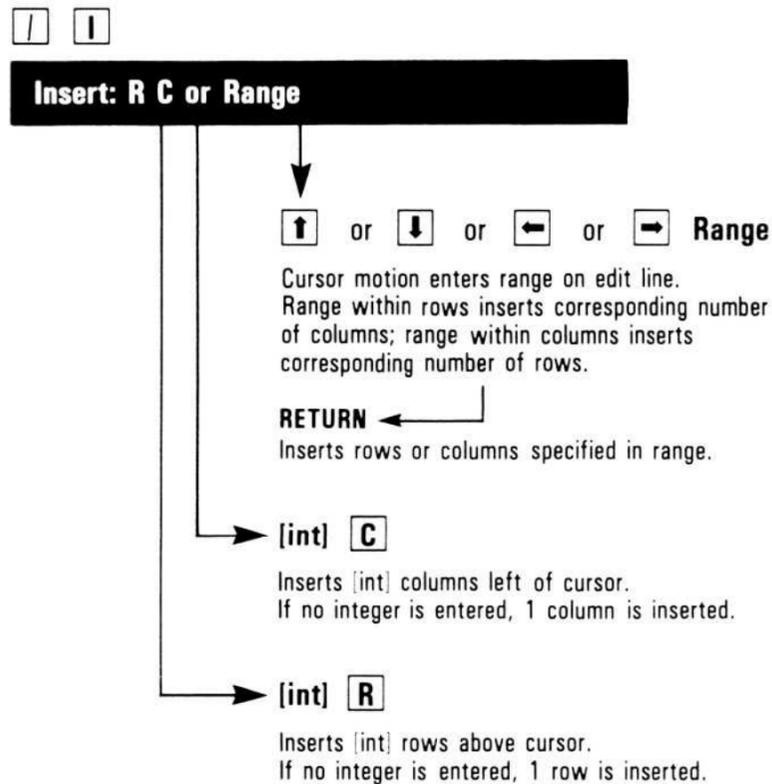
**NOTE:** If you want a question-mark (?) in a label on your worksheet, type CONTROL-Q. A "?" appears in the label where you typed CONTROL-Q.

## Help Facility Options

Options available in the Help Facility are listed at the bottom of all screens of Help text. Options are executed by typing one key. Not all options are available with every Help screen. All the Help Facility options and the keys that execute them are listed below.

<b>Option Key</b>	<b>Option</b>	<b>What it Does</b>
N	NEXT	Displays the next screen of Help text.
P	PREVIOUS	Displays previous screen of Help text.
T	TOPICS	Displays a list of available Help topics. You use ↑ and ↓ to move the cursor up and down the list; press RETURN to select the topic you need help with.
ESCAPE SHEET		Returns you to where you were on the worksheet when you asked for help.
CONTROL-C CANCEL		Cancels the command you were in when you asked for help. If you were not in a command, you are returned to the worksheet.
RETURN SELECT		Selects the help topic on which the cursor rests in the list of topics.
ARROWS: CURSOR		↓ moves the cursor down the list of topics; ↑ moves the cursor up the list of topics.
? HELP		Gives information on how to use the Help facility.

## INSERT COMMAND—/I



The Insert command adds blank rows or columns to the worksheet.

In response to `/I`, the prompt line reads `Insert: R C or Range`. The `R` and `C` are options:

- `R`     Insert Rows.
- `C`     Insert Columns.

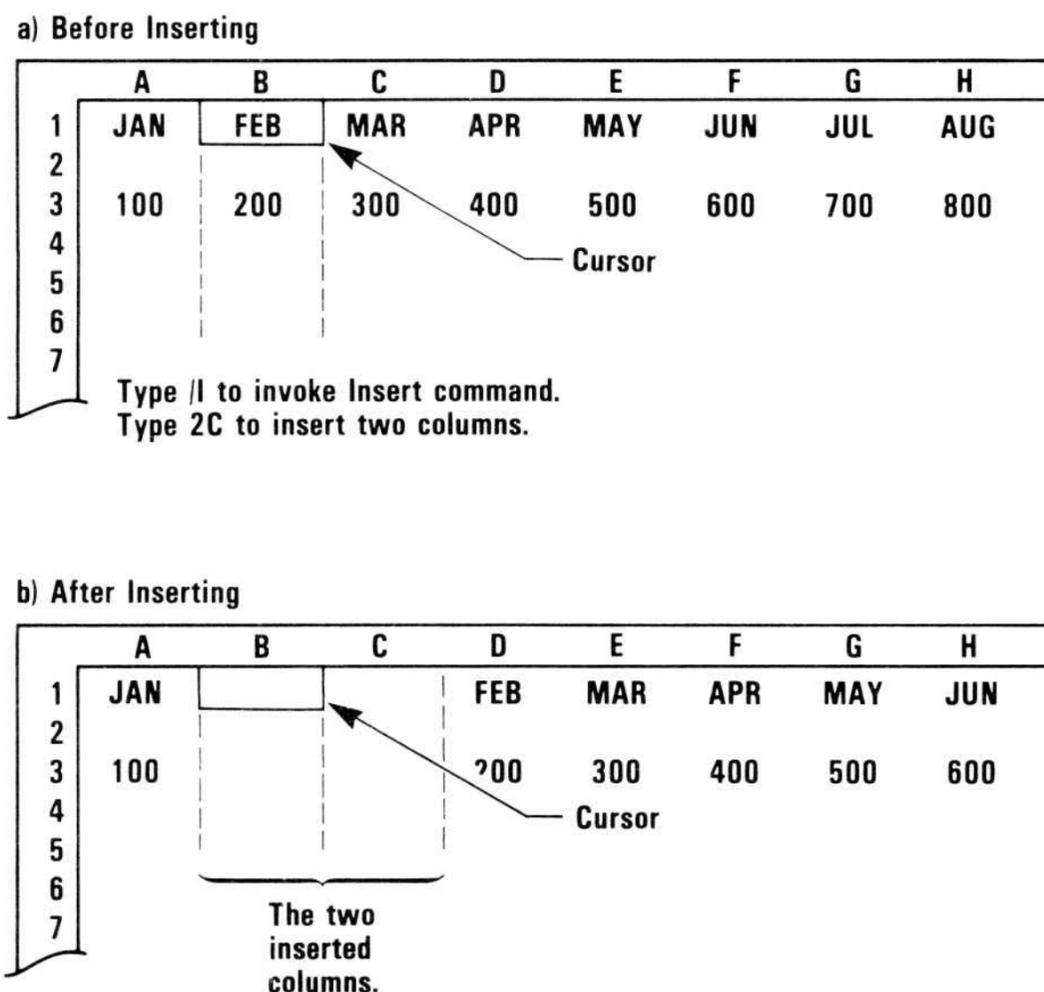
As the prompt indicates, there are two ways to insert rows or columns: (1) by specifying the number of rows or columns to be inserted, or (2) by indicating a range where the rows or columns are to be inserted.

To specify the number of rows to insert, type `/I(integer)R`. The integer you type is the number of rows that are inserted. Rows are inserted starting with the row that contains the cursor.

To specify the number of columns, type `/I(integer)C`. The integer you type is the number of columns that are inserted. Columns are inserted starting with the column that contains the cursor.

To insert rows or columns by indicating a range, type `/I` and point to the range with the cursor or type the beginning and ending coordinates. Ranges within a column indicate that rows are to be inserted; the range `A1...A3`, for example, inserts 3 rows (rows 1, 2, and 3). Ranges within a row insert columns; for example, the range `A1...C1` inserts 3 columns (columns A, B, and C).

Figure 3-13 shows how columns are inserted. Inserting rows is similar.



010-022

Figure 3-13. Inserting Columns

If no integer precedes the R or C, one row or column is inserted.

One cell cannot be inserted, only an entire row or column.

When rows or columns are inserted into the range of an arithmetic function, the range is expanded to include the insertion. (Rows or columns cannot be inserted before the first location or after the last location of the range; the point of insertion must lie between the first and last locations.)

**Note:** Because rows move down and columns move to the right to make space for an insertion, a row cannot be inserted if row 254 contains an entry; a column cannot be inserted if column BK contains an entry. To make an insertion, you must erase the contents of these cells (with the Blank command) or move them (with the Move command). You must then save the worksheet, clear the worksheet with /CAY, and then reload the worksheet.

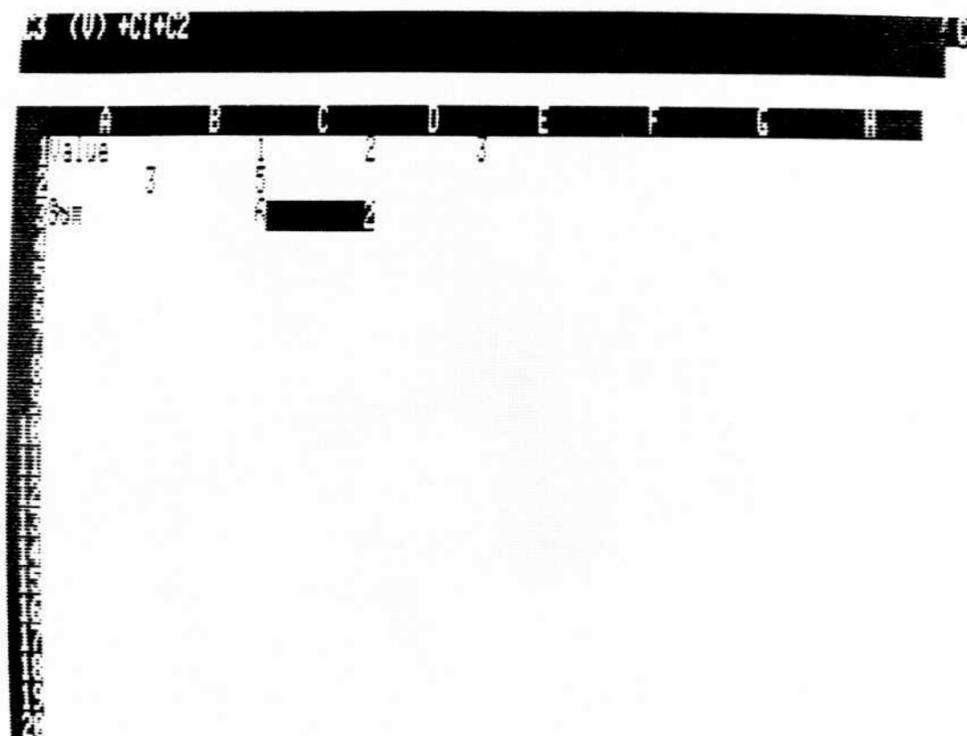
## Row Option — /IR

The Row option inserts blank rows. All rows from the first row in the insert range to the bottom of the worksheet move down to make room for the new rows.

All cell references in formulas on the worksheet are changed to correspond to the new coordinates of the rows that moved. For example, if a formula contains the coordinate C2 and one row is inserted above row 2, the coordinate in the formula is changed to C3.

Example:

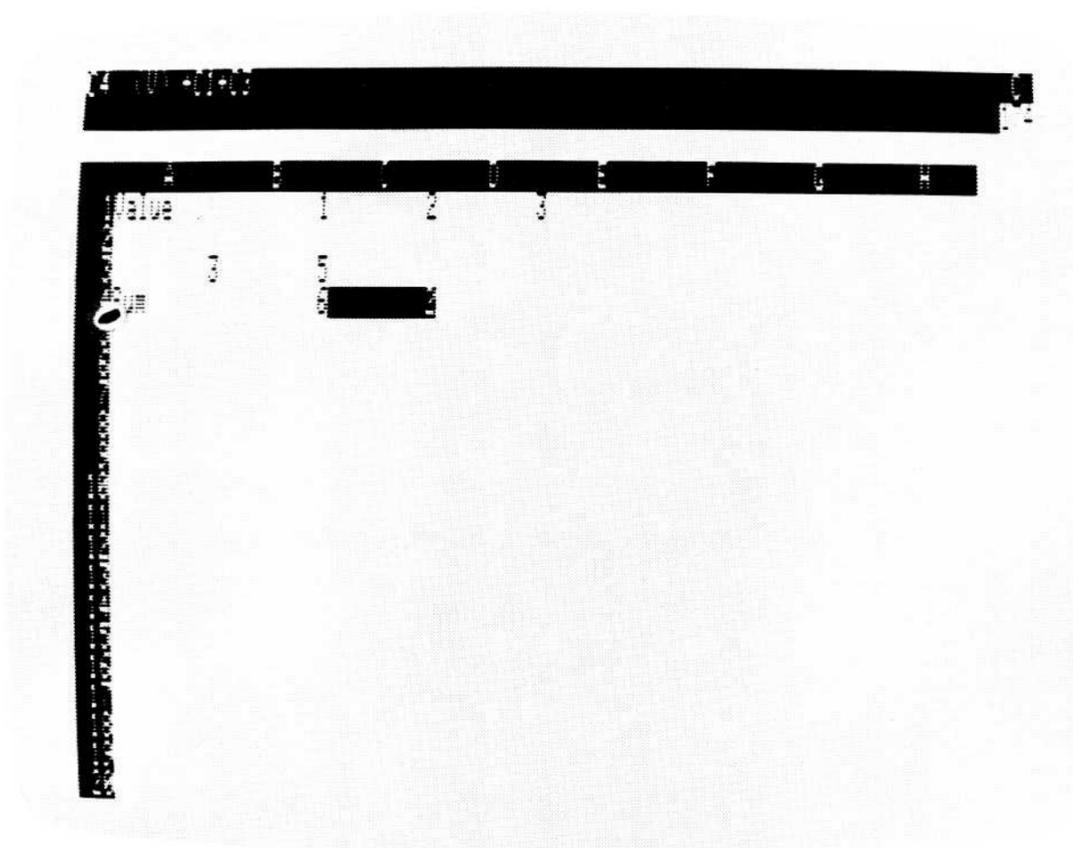
- | <b>Type:</b>  | <b>Result:</b>  |
|---|---|
| 1. /CAY   | The worksheet clears and the cursor moves to A1.                      |
| 2. Value → 1 → 2 → 3<br>RETURN > A2 RETURN<br>Formula → +B1+C1 →<br>+C1+D1 RETURN > A3<br>RETURN Sum →<br>+B1+B2 → +C1+C2<br>RETURN | Entry Line: C3 (V) +C1+C2<br>The screen should look like figure 3-14. |



010-030/P

Figure 3-14. Worksheet Before Inserting Row

- 3. ↑ Cursor moves up to C2.
- 4. /IR Row 2 moves down (becoming row 3), creating a new blank row 2.
- 5. ↓↓ Entry Line: C4 (V) + C1 + C3  
The cell reference in the formula has changed from +C1+C2 to +C1+C3 because of the inserted row. The screen should look like Figure 3-15.



010-031/P

Figure 3-15. Worksheet After Inserting Row

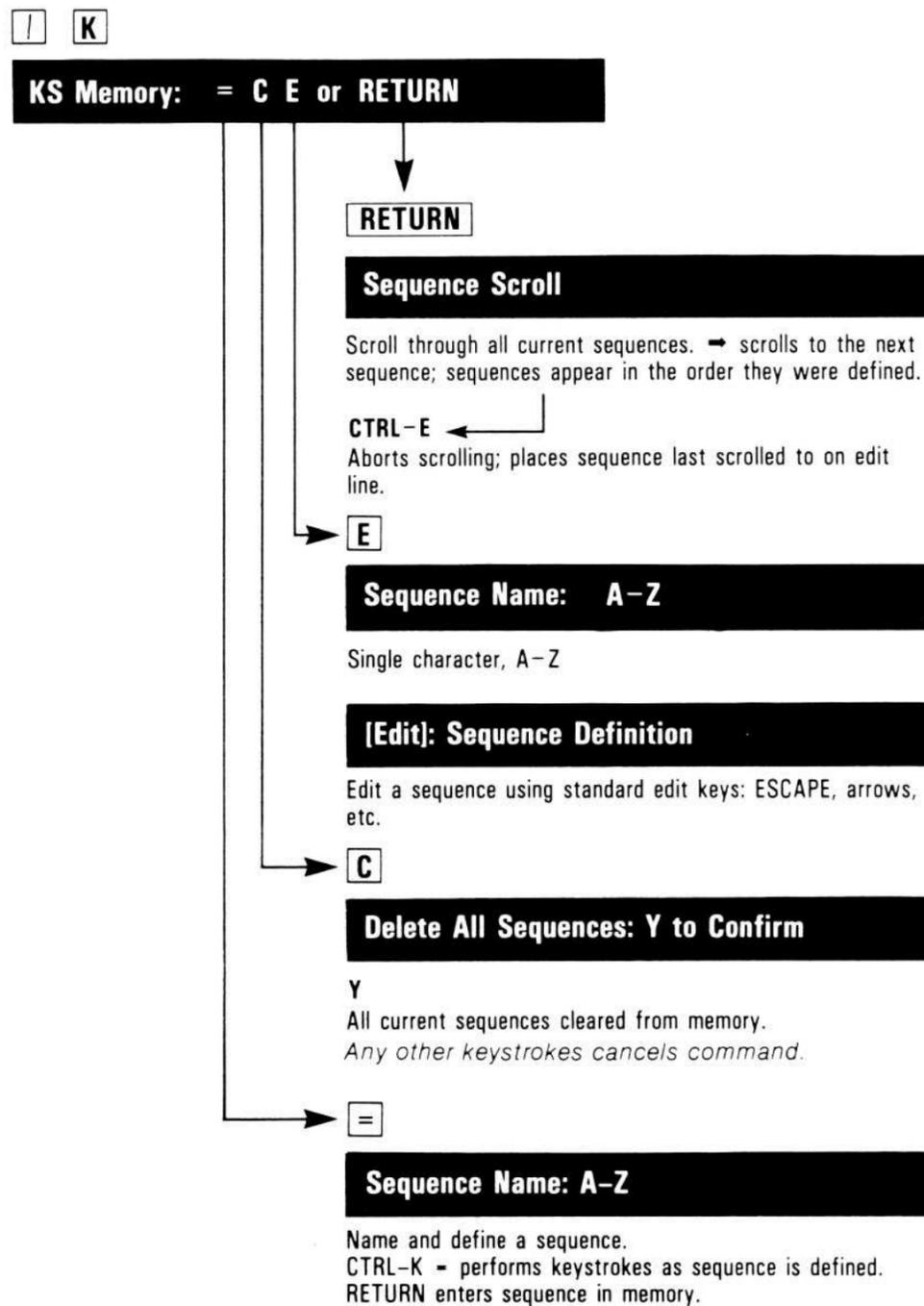
- 6. ↑↑ /DRY ↓ Entry Line: C3 (V) + C1 + C2 The new row is deleted, and the formula returns to its original value.

### Column Option – /IC

The Column option inserts blank columns. All columns from the first column in the insert range to the right edge of the worksheet move to the right to make room for the new columns.

All cell references in formulas on the worksheet are changed to correspond to the new coordinates of the columns that moved. For example, if a formula contains the coordinate C2 and a column is inserted to the left of column C, the coordinate is changed to D2.

## KEYSTROKE MEMORY COMMAND—/K

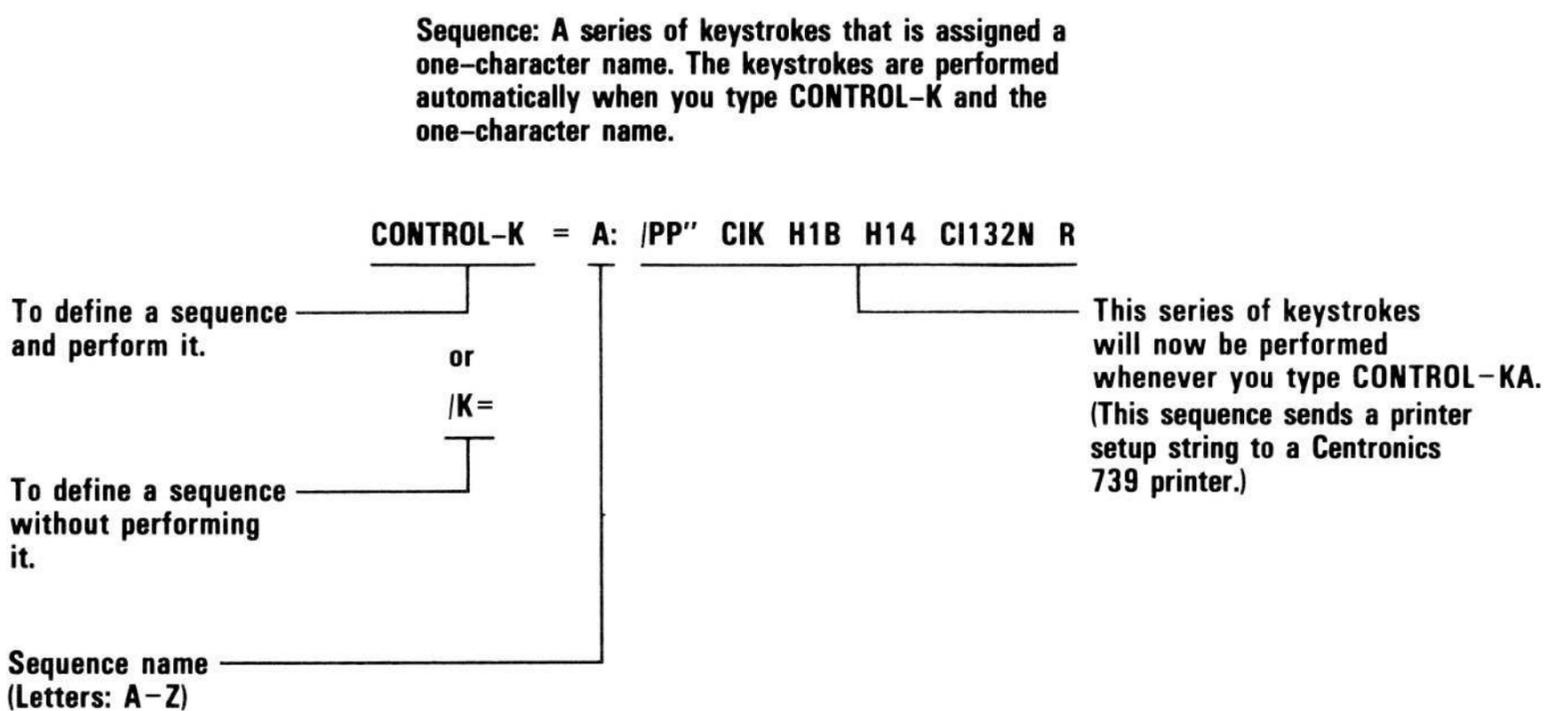


The Keystroke Memory command lets you create a one-character abbreviation for keystroke sequences. Instead of re-typing a long series of keystrokes, /K captures the keystrokes in memory and lets you give them a one-character name. Then, whenever you want to perform that sequence, you type CONTROL-K followed by the one-character name you assigned, and the VisiCalc program does the keystrokes for you. You can capture in memory 125 keystrokes. This feature of the VisiCalc program is a great time-saver for building worksheets and doing complex, repetitive tasks.

The command has four options:

- =** Define a sequence
- C** Delete all sequences
- E** Edit a sequence
- RETURN** Review all sequences

The following figure illustrates how the Keystroke Memory command works.



**Examples:**

Sequence Definition	Name	Strings
CONTROL-K =	A:	/PP" ^ CIK ^ H1B ^ H14 ^ CI132N ^ R
/K =	B:	JAN ^ > FEB ^ > MAR ^ > APR ^ > MAY ^ R
/K =	C:	/R: ^ > . ^ > ^ > ^ > ^ R

010-023

**Figure 3-16. Defining a Keystroke Sequence**

## The Define Sequence Option—/K= or CONTROL-K=

The Define Sequence option lets you determine the series of keystrokes you want to put into memory for using later and lets you assign the series a one-character name. You can define sequences that enter data, move to other cells, and use several commands to consolidate worksheets. The only valid sequence names are the letters A-Z.

There are two ways to define a sequence: /K=(sequence name) and CONTROL-K=(sequence name). CONTROL-K= allows you to define a sequence as you enter it on the worksheet; the keystrokes you type are executed as normal VisiCalc keystrokes *and* saved as a sequence. This allows you to try out different keystroke combinations before storing a sequence in memory. /K= captures the keystrokes you type without executing them.

A sequence can be defined at any time. After it has been defined and stored in memory, a sequence is invoked by typing CONTROL-K followed by its one-letter name. The only valid sequence names are the letters A-Z. Typing an integer after you type CONTROL-K but before you type a sequence name will execute the sequence that number of times. For example, CONTROL-K 5 B means you will execute the sequence named "B" 5 times.

The maximum size of a sequence is 123 characters. Because some keystrokes are represented within sequences as more than one character, you might not be able to fit 123 keystrokes in a single sequence. Using the chain-to-another-sequence character (^K) in a sequence allows you to execute multiple sequences. With this feature, you can overcome any limitation (except memory size) on the length of a sequence. You can split a long sequence into two shorter sequences, and chain the second to the first, to get the same effect. You would chain the "A" sequence to your first sequence with a "^KA". You can refer to a sequence(s) from within another sequence. The primary sequence will execute the imbedded sequence(s) and then return to finish its own series of commands. You can also type an integer after ^K and before the name of the sequence you are chaining. For example, typing "^K5A" at the end of a macro loads the sequence named A and executes it 5 times.

**NOTE:** A sequence can be used to load new worksheets. Make sure, however, that the new worksheets do not contain any sequence names used in the previous worksheet. If the first worksheet uses **A**, **B** and **C**, the second worksheet should not use these letters as sequence titles.

When defining a sequence using CONTROL-K = , keystrokes that cannot be represented with one character, such as cursor movements and RETURN, are translated into special characters in memory. If you edit a sequence (with /KE) that was defined by CONTROL-K = , keystrokes such as RETURN and the arrow keys are translated into these special characters.

K will be displayed in the far right corner of the Entry Line while a sequence is being defined with CONTROL-K = .

When defining a sequence using /K = , you must type special characters for RETURN and arrow keys. You must also do so when editing sequences with the /KE option. The following table shows the special characters and the keystrokes they represent.

<b>Characters</b>	<b>What They Represent</b>
^^	Caret (^)
^U	Up arrow (↑)
^D	Down arrow (↓)
^<	Left arrow (←)
^>	Right arrow (→)
^?	Help
^TF	Tab forward to next cell with tab set
^TB	Tab backward to last cell with tab set
^R	RETURN
^B	ESCAPE (delete)
^C	CONTROL-C (break)
^P	Pause and display message during a sequence
^K	Chain current sequence to another sequence

The following example shows how to define a sequence using CONTROL-K = (sequence name).

Example:

Type:	Result:
1. /CAY	Entry Line: [A1]
2. CONTROL-K	Prompt Line: KS MEMORY; NAME # =
3. =	<p>You have chosen to interactively define a sequence; the other choice, NAME, executes a sequence.</p> <p>Prompt Line: SEQUENCE NAME: A-Z</p> <p>The program is expecting a single letter to use as the sequence name for the keystrokes you are about to enter.</p>
4. A	<p>Notice the K in the upper-right corner of the Entry Line; while you are defining a sequence using CONTROL-K = , this K will be displayed.</p>
5. JAN → FEB → MAR → APR → MAY → JUN → JUL → AUG → SEP → OCT → NOV → DEC RETURN CONTROL-K	<p>Labels entered on worksheet as you type them. Pressing CONTROL-K at the end enters the sequence into memory.</p>
6. CONTROL-K A	<p>Prompt and Edit Lines flash as sequence is executed.</p>

Here is the same example using the /K= option:

<b>Type:</b>	<b>Result:</b>
1. /CSY	Entry Line: [A1]
2. /K=M	Prompt Line: SEQUENCE DEFINITION Edit Line: M:  You are going to define a sequence that enters column labels for a monthly budget, JAN – DEC.
3. JAN^>FEB^>MAR^> APR^>MAY^>JUN^> JUL^>AUG^>SEPT^> OCT^>NOV^>DEC^ R RETURN	Edit Line: Reflects keystrokes as you type them. Pressing RETURN enters the sequence into memory.
4. CONTROL-K M	Prompt and Edit Lines flash as though you were typing the keystrokes very quickly. The column labels are entered on your worksheet in a fraction of the time it would have taken you to do it manually.
5. ↓CONTROL-K M	The sequence is performed again. The month labels appear in columns L-W.

You can delete a specific sequence using the Define Sequence option. Just type /K= followed by the sequence name. This prompt will appear:

SEQUENCE EXISTS. Y TO REPLACE.

Type **Y**, indicating that you want to replace the old sequence, then press **RETURN**. Because the most recently defined sequence supersedes old sequences of the same name, this re-defines the sequence as a sequence which does nothing.

All sequences that you defined for a worksheet are saved with the worksheet. Sequences are cleared, along with printer settings, attributes, and formats, with the /CA option of the Clear command.

## USING PAUSES (^P) IN SEQUENCES

A pause in a sequence lets you display a message in the middle of a sequence. The message is displayed on the edit line until you type RETURN. After you type RETURN, the sequence continues.

Pauses are useful in a file-save sequence to display a prompt that tells you when to change diskettes, in a print sequence to ask if your printer is properly set up, and in other sequences.

Pauses in a sequence are specified when the sequence is being defined. If you are defining a sequence using CONTROL-K = (sequence name), you specify a pause by following these steps:

1. Type CONTROL-P. This indicates that you want a pause.
2. Type the message you want to display during the pause and end it with a colon (:). Use ESCAPE to change mistakes. The message is entered as a label when the sequence is defined, but is displayed on the prompt line until you press RETURN when the sequence is performed.
3. Continue defining the sequence.

Your message and (PRESS RETURN) are displayed when the pause is encountered in the sequence and the sequence stops. After you press RETURN, the sequence continues.

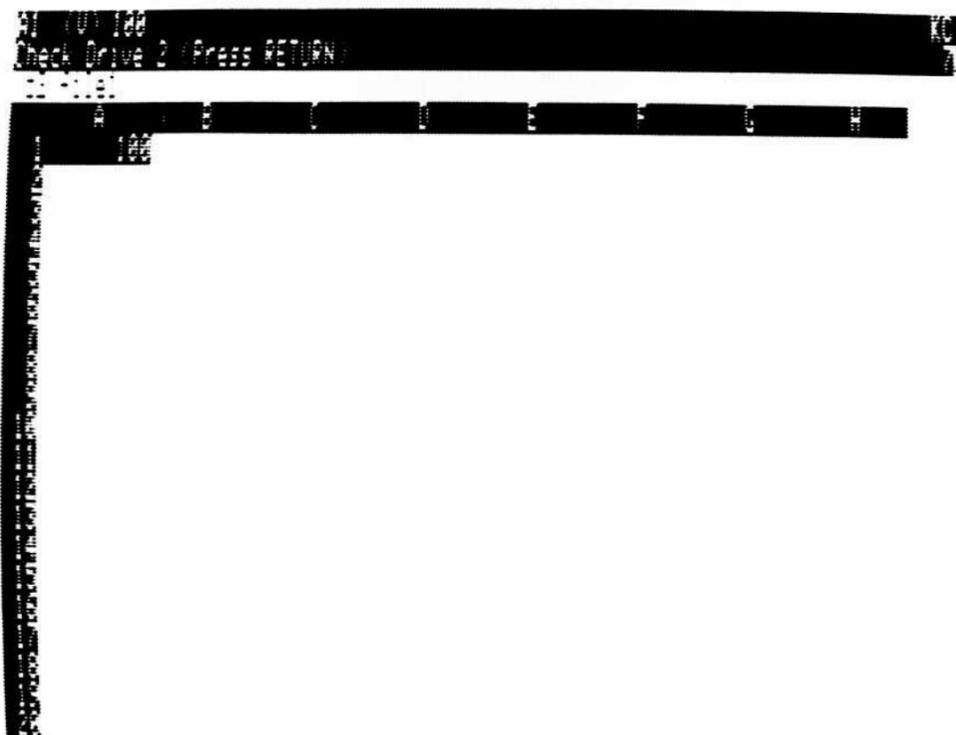
The message displayed during a pause is a reminder to you, not an instruction to the VisiCalc program. The program does not check to see if an action called for in a prompt has been done. For instance, consider the case of a sequence that saves a file onto diskette. If this sequence contains a pause that asks you to switch diskettes in drive 1, the program does not check to see if you have actually switched diskettes.

To put a pause in a sequence being defined with /K = , follow these steps (1) type ^P - this indicates that you want a pause; (2) type the message you want displayed on the prompt line during the pause; (3) type a colon (:) to indicate the end of the message.

Here is an example of how to use a pause in a sequence.

Example:

<b>Type:</b>	<b>Result:</b>
1. <b>/CAY</b>	Screen clears and worksheet is reset; cursor goes to <b>A1</b> .
2. <b>100 RETURN</b>	Entry Line: <b>A1 (V) 100</b>
3. <b>/K=A</b>	You're going to define a sequence called <b>A</b> .
4. <b>/SS.D2/FILE1</b>	The keystrokes you type appear on the Edit Line. So far, sequence <b>A</b> saves a worksheet onto the diskette in drive 2 under the name <b>FILE1</b> .
5. <b>^P</b>	Indicates that you want a pause in the sequence.
6. <b>CHECK DRIVE 2:</b>	This is the message that will appear during the pause. You must end the message with a colon (:).
7. <b>^R RETURN</b>	The last keystroke in the sequence is <b>RETURN</b> , which is represented by <b>^R</b> on the Edit Line. Pressing <b>RETURN</b> after typing <b>^R</b> puts the sequence in memory.
8. <b>CONTROL-K A</b>	Keystrokes in sequence <b>A</b> are performed until the pause is encountered; then your screen should look like the following Figure 3-17.



010-032/P

Figure 3-17. A Pause in a Sequence

### The Clear Option—/KC

The Clear option irretrievably clears *all* current sequences from memory. After you type /KC, this prompt appears on the prompt line: DELETE ALL SEQUENCES. Y TO CONFIRM. Typing Y deletes all sequences currently in memory; typing anything other than Y cancels the command and returns you to the worksheet.

### The Edit Option—/KE

The Edit option prompts for a sequence name and places the sequence you specify on the edit line. You can then edit the sequence, using the normal edit keys (ESCAPE, →, ←, ↑, ↓); you can also change the sequence name. When you are finished editing a sequence, press **RETURN**. The edited sequence will then be stored in memory and can be invoked at any time. Keystroke Memory sequences are saved with worksheets.

The following example uses the /KE option.

Example:

<b>Type:</b>	<b>Result:</b>
1. /CAY	The worksheet clears, the cursor moves to A1, and the column width is set to 9.
2. /K	Prompt Line: KS Memory: = C E or RETURN
3. =	Prompt Line: SEQUENCE NAME: A-Z Define Sequence option of the Keystroke Memory command.
4. A	Prompt Line: SEQUENCE DEFINITION Edit Line: A:
5. 100000^R/R: ^>.^>^>^>^R RETURN	Edit Line shows keystroke sequence as you type it. Pressing RETURN puts the sequence in memory. This sequence enters the value 100,000 at A1 and replicates it into B1 through D1.
6. CONTROL-K	Prompt Line: KS Memory: Name # =
7. A	Prompt Line: flashes prompts that correspond to keystrokes in sequence.
8. ↓/K	Prompt Line: KS Memory: = C E or RETURN
9. E	Prompt Line: Sequence Name: A-Z
10. A	Prompt Line: [Edit]: Sequence Definition Edit Line: A:100000^R/R:^>.^>^>^>^R
11. →→2	Edit Line: A:100000^R/R:^>.^>^>^>^R
12. RETURN	The sequence is altered: now the number 2100000 will be entered and replicated.
13. CONTROL-K	Prompt Line: KS Memory: NAME # =
14. A	Prompt Line: Flashes prompts that correspond to keystrokes in sequence; 2,100,000 is entered and replicated.

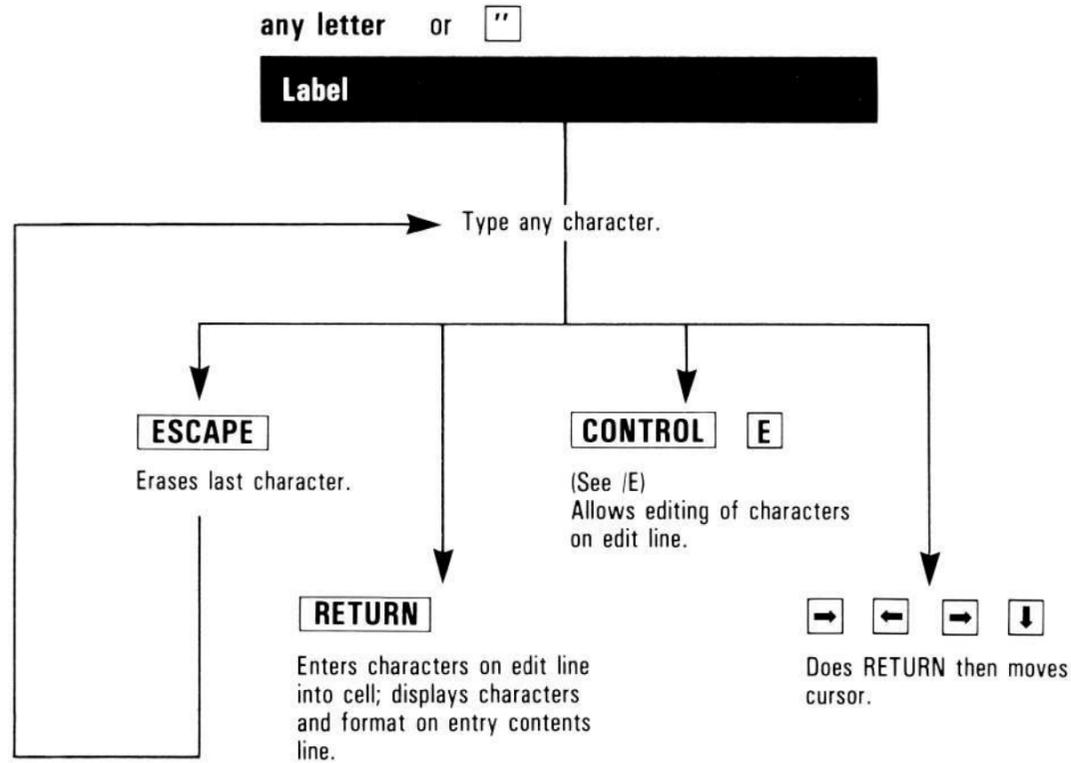
You can delete a specific sequence using the Edit option. Just type **/KE** followed by the sequence name. Then delete all the characters in the sequence, but not the sequence name, using **ESCAPE**. It is cleared from memory.

### **The Review Sequences Option—/K RETURN**

The Review Sequences option scrolls through all the current sequences. After you type **/K RETURN**, the first sequence and its keystrokes will appear on the Edit Line. Typing **→** places the next sequence on the Edit Line. Sequences appear in the order they were created.

If you type **CONTROL-E** while the sequence is on the edit line, **[EDIT]** will appear on the prompt line. You can use the regular edit keys (**ESCAPE**, **→**, **←**, etc.) to edit the sequence.

## LABELS



A label is an entry not intended to be used in calculations; it usually gives some information about the entries on the worksheet. A label can have any keyboard character.

When the prompt and edit lines are clear, the VisiCalc program looks at the first character typed to determine whether the entry is a label or a value. If the character is not a number or arithmetic operator, the entry is a label. The prompt line reads `Label` and the edit line contains the character that was typed followed by the edit cue. As more characters are typed, they appear on the edit line.

The label is displayed in the cell as it is typed; it is not entered until RETURN or an arrow key is pressed. If any other characters are in that cell, the label replaces them. If the label is longer than the column width, it is truncated on the cell display, but the full label is kept in memory. When the cursor is in a cell that contains a label, the entry line contains the coordinate, followed by (L) and the full label.

To display a column of long labels, use /GCC, the Single Column Width of the Global command, to make the column wider.

To type a question mark (?) in a label, type CONTROL-Q. The ? on your keyboard always displays Help Facility information on your screen.

If you make an error while typing a label, type ESCAPE to erase the error and continue typing. If you want to correct an error in an entry after it has been entered, use the Edit command (/E).

If a formula references a cell that contains a label, the reference has a value of 0.

Example:

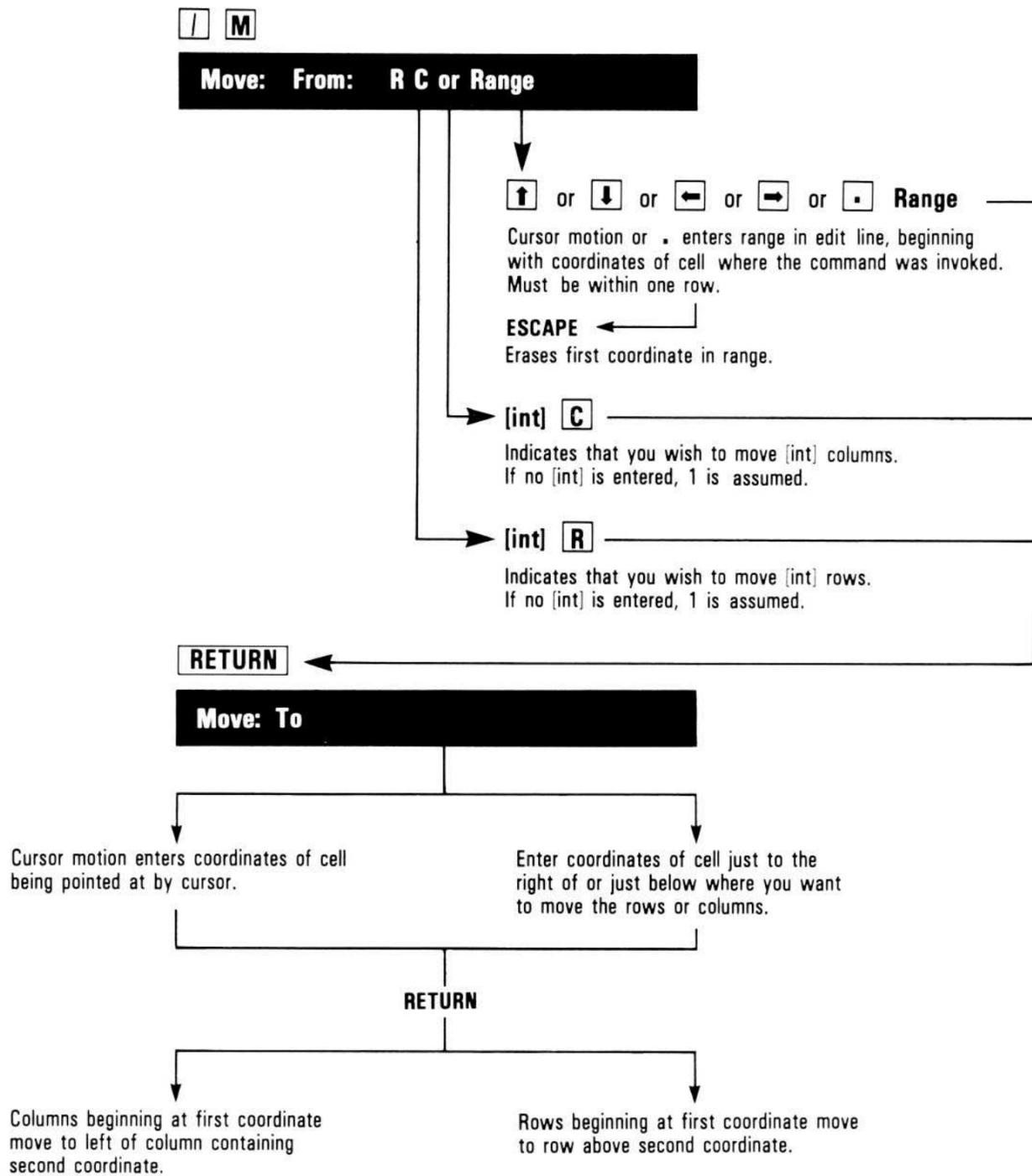
<b>Type:</b>	<b>Result:</b>
1. >A1 RETURN	The cursor moves to A1.
2. P	Entry Line: A1 Prompt Line: Label Edit Line: P A1: P
3. eriod	Prompt Line: Label Edit Line: Period A1: Period
4. RETURN	Entry Line: A1 (L) Period Prompt Line: clear Edit Line: clear A1: Period
	If you press an arrow key to end the label, the cursor moves to the next location in the direction of the arrow.

Type `"` to start a label with a character that normally begins a value—a number, arithmetic operator, `(`, or `@`. The `"` is invisible and merely tells the VisiCalc program that the entry is a label. The Mode option of the Attribute command (`/AM`) can be used to specify that a cell may only contain labels; see the section on "The Attribute Command" for more information.

Example:

<b>Type:</b>	<b>Result:</b>
1. <code>&gt;B1 RETURN</code>	The cursor moves to B1.
2. <code>"</code>	Prompt Line: Label Edit Line: clear
3. <code>.575*B2</code>	Prompt Line: Label Edit Line: <code>.575*B2</code> B1: <code>.575*B2</code>
4. <code>RETURN</code>	Entry Line: B1 [L] <code>.575*B2</code> Prompt Line: clear Edit Line: clear B1: <code>.575*B2</code>  The [L] on the entry line identifies the entry as a label. If the entry were a value, the entry line would contain [V] and B1 would contain the calculated value of the formula.

**MOVE COMMAND—/M**



The Move command moves rows or columns to another position on the worksheet. All cell references in formulas are changed to correspond to the new coordinates that result when the rows or columns are moved.

The Move command does the following:

1. Removes one or more rows or columns from their place on the worksheet.
2. Closes the space left vacant by moving rows upward or columns to the left.
3. Opens the space needed for inserting the removed rows by moving rows downward; opens space for removed columns by moving columns to the left.
4. Inserts the removed rows in the opened space.

In response to /M, the prompt line reads Move: From: R C or Range.

Moving rows and columns takes two steps: (1) specifying the rows or columns you want to move, and (2) indicating where you want them moved to.

### **Specifying the Rows or Columns to be Moved**

After typing /M, you can specify which rows or columns you want to move two ways: by indicating a range, or by typing an integer and R or C.

You can specify a range of rows or columns to be moved by moving the cursor or typing the range coordinate. The coordinate of the cell where the command was invoked is the first coordinate in the range. Type a period (.) to display the ellipsis (...); the last coordinate in the origin range can now be specified either by typing it or moving the cursor. Ranges within a column indicate that rows are to be moved; the range A1...A3, for example, specifies that 3 rows are to be moved. Ranges within a row indicate that columns are to be moved; for example, the range A1...C1 specifies that 3 columns are to be moved.

After typing /M and moving the cursor or typing ., you can use ESCAPE to erase the coordinate on the edit line and either type the coordinates of the first row or column you want to move, or point to it with the cursor. You can then press . and type a coordinate or move the cursor to indicate the last row or column in the source range.

Alternatively, you can specify that you want to move a number of rows or columns by typing the number and R or C. /M4C, for example, says that the column where the cursor is and the three columns to the right are to be moved; /M4R says that the row where the cursor is and the three rows below it are to be moved.

## Indicating Where the Rows and Columns Are to be Moved

After you type a number and R or C, you must indicate the first row or column in the destination range, either by pointing to it or by typing it.

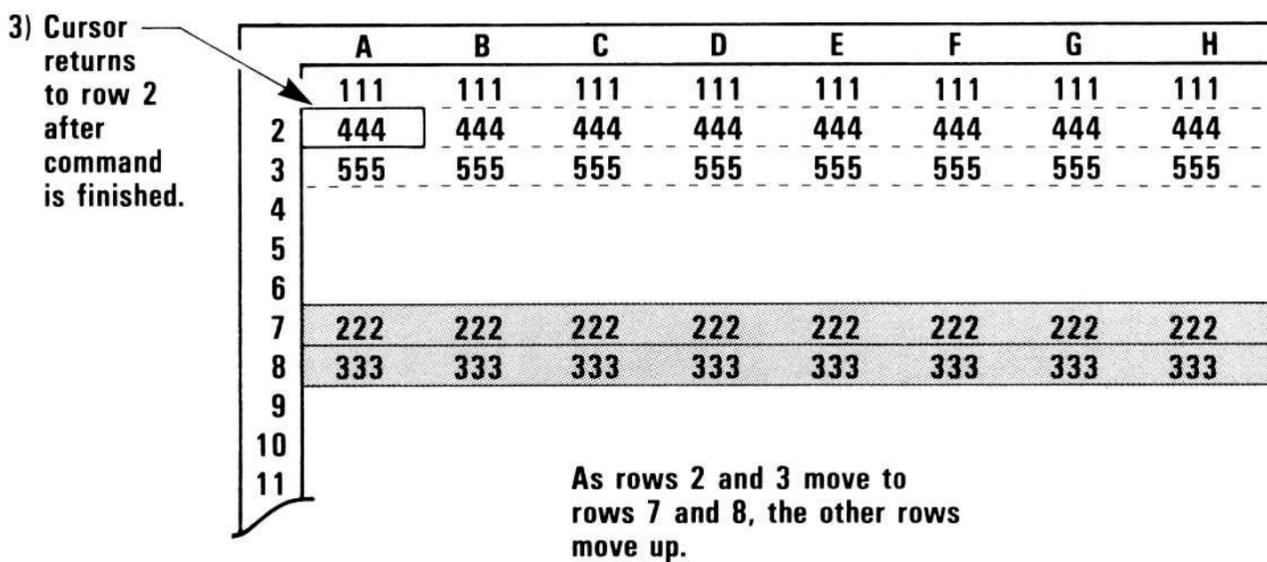
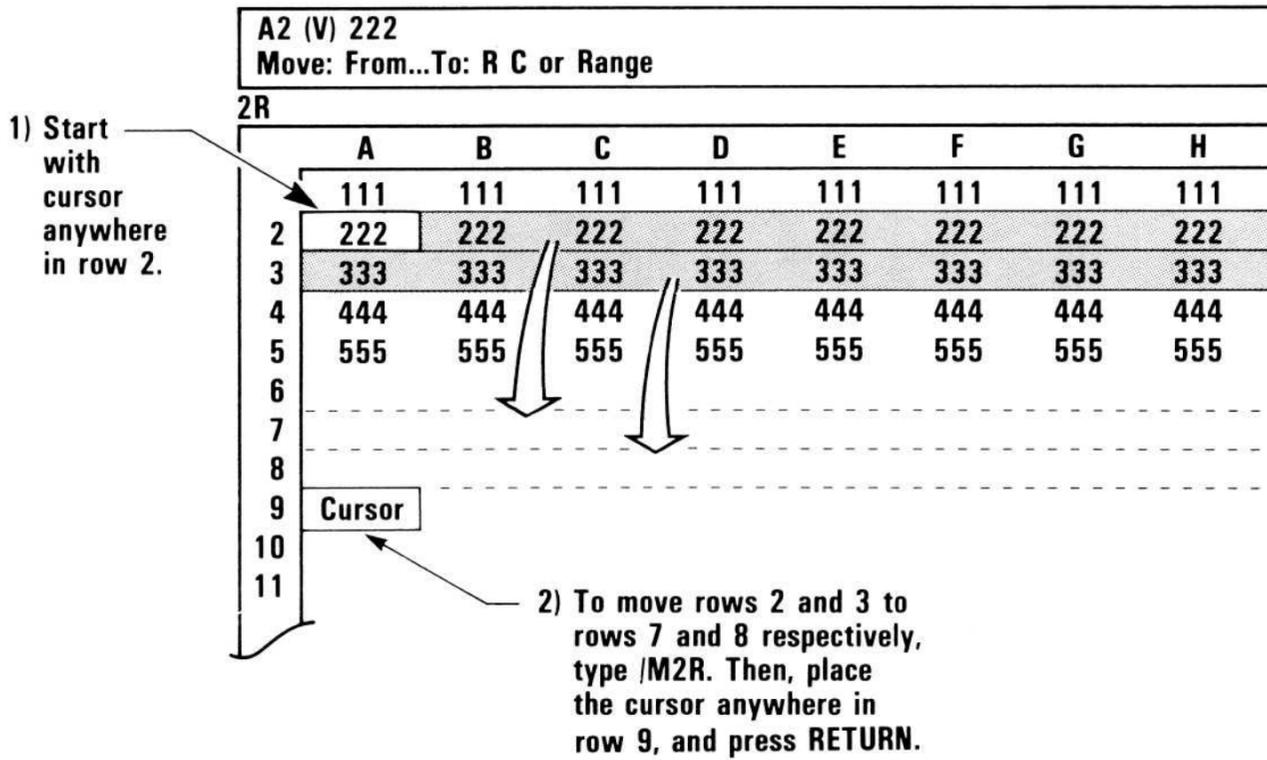
To complete the command, press RETURN.

Moving a row or column can create a forward reference or circular reference, requiring extra manual recalculations (!) or preventing the VisiCalc program from yielding accurate results. See "Order of Recalculation" (/GO) in the "Values" section of this chapter and "Resolving Forward References" in Lesson Six for details and information on how to avoid circular references.

## Moving Rows

The Move command moves all rows below the origin up to fill the gap; then it inserts the rows by moving the destination and all rows below it down to create a gap. Placing the cursor while moving rows downward is different from moving rows upward. When you move rows down, specify the destination as the row *below* the last row you want the rows moved to; when you move a row up, specify the destination as the first row you want the rows moved to.

Figure 3-18 shows where to position the cursor when moving rows.



010-024

**Figure 3-18. Moving Two Rows Downward**

After the row is moved, the cursor returns to its original location.

Figure 3-19 shows where to place the cursor when moving rows up the worksheet.

1) Before moving rows upward.

	A	B	C	D	E	F	G	H
1	111	111	111	111	111	111	111	111
2	444	444	444	444	444	444	444	444
3	555	555	555	555	555	555	555	555
4								
5								
6								
7	222	222	222	222	222	222	222	222
8	333	333	333	333	333	333	333	333
9								
10								
11								

	A	B	C	D	E	F	G	H
1	111	111	111	111	111	111	111	111
2	Cursor							
3								
4	444	444	444	444	444	444	444	444
5	555	555	555	555	555	555	555	555
6								
7	222	222	222	222	222	222	222	222
8	333	333	333	333	333	333	333	333
9								
10								
11								

2) To move rows 7 and 8 upward to rows 2 and 3, respectively, type /M2R. Then, place the cursor in row 2 and press RETURN.

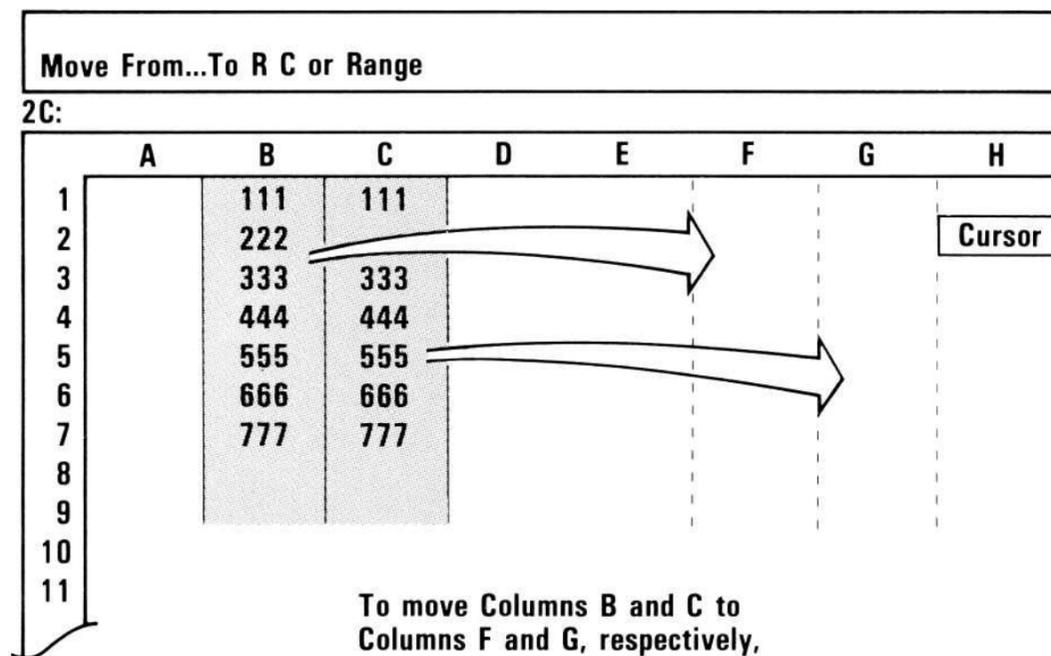
010-025

Figure 3-19. Moving Two Rows Upward

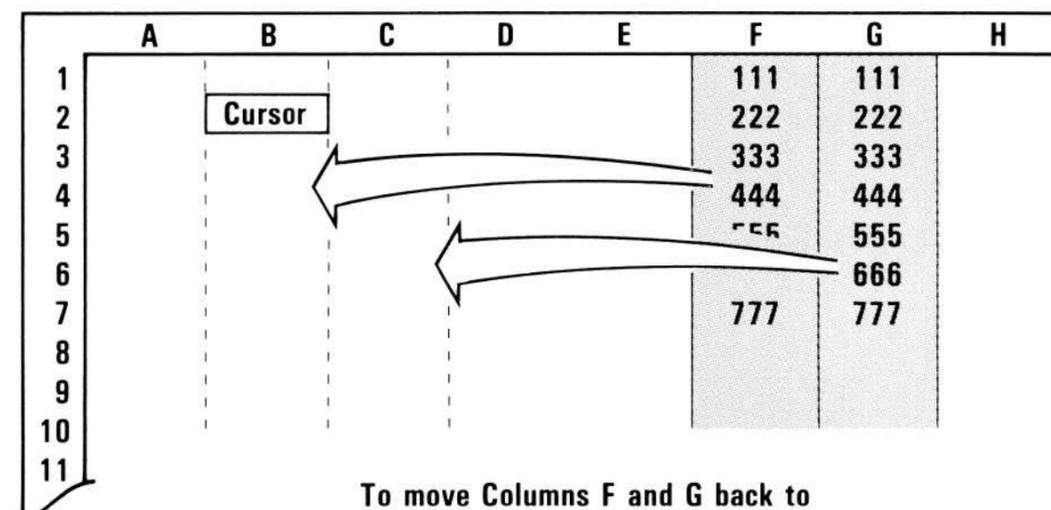
### Moving a Column

The Move command moves all columns to the right of the origin left to fill the gap; then it inserts the columns by moving the destination and all columns to the right of it to the right to create a gap. Moving columns to the right is different from moving columns to the left. When you move columns to the right, specify the destination as the column *to the right of* the last column you want the columns moved to; when you move a column to the left, specify the destination as the last column you want the columns moved to.

Figure 3-20 shows how to move columns.



To move Columns B and C to Columns F and G, respectively, type /M2C. Then, place the cursor anywhere in Column H and press RETURN.



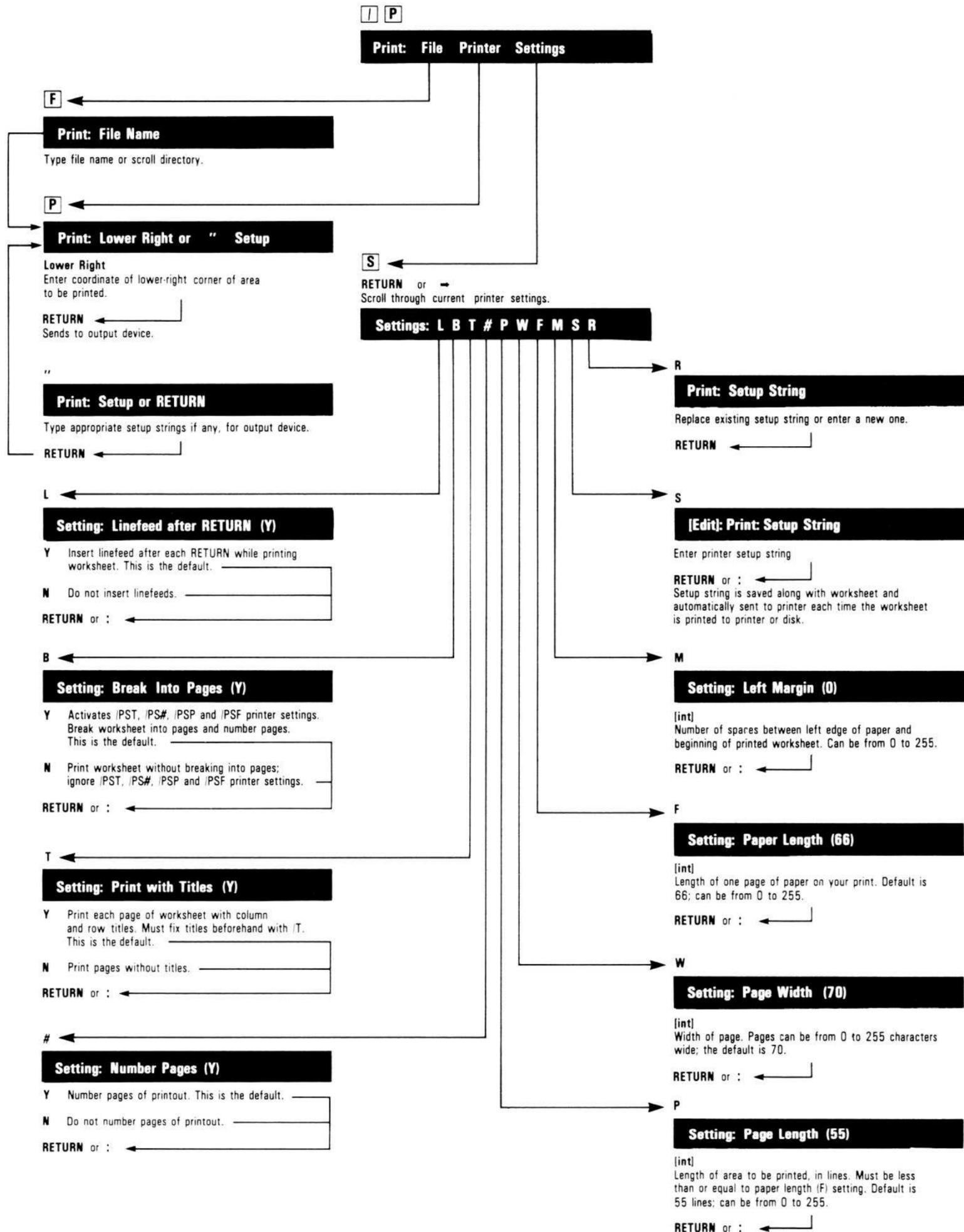
To move Columns F and G back to Columns B and C, respectively, type /M2C. Then, place the cursor anywhere in Column B and press RETURN.

010-026

Figure 3-20. Moving Two Columns

After the column is moved, the cursor returns to its original location.

PRINT COMMAND — /P



The Print command produces an image of the worksheet just as it appears on the screen. This image can be sent to one of several destinations:

- A printer.
- A diskette file.

You can print the normal worksheet with numerical entries or with your formulas on it. Use the Expression Format option of the Attribute command (/GAE) with the Print command to print the formulas behind your worksheet. See the following section on "Printing Worksheet Formulas" for details.

### **Printing on the Printer—/PP**

The Printer option of the Print command (/PP) sends the image of the worksheet to a printer. The upper-left corner of the rectangle to be printed is the location of the cursor when the command is entered; the program prompts you to enter the lower-right coordinate.

If the screen is split when the command is entered, the windows are ignored; only the rectangle specified by the upper-left and lower-right coordinates is printed. Use the Go To command (>) if you want to move the cursor to an upper-left location that is fixed as a title and inaccessible with the arrow keys.

Typing CONTROL-C stops the printer.

The following example prints the area of the worksheet bounded by A1 on the upper-left and J14 on the lower-right.

Example:

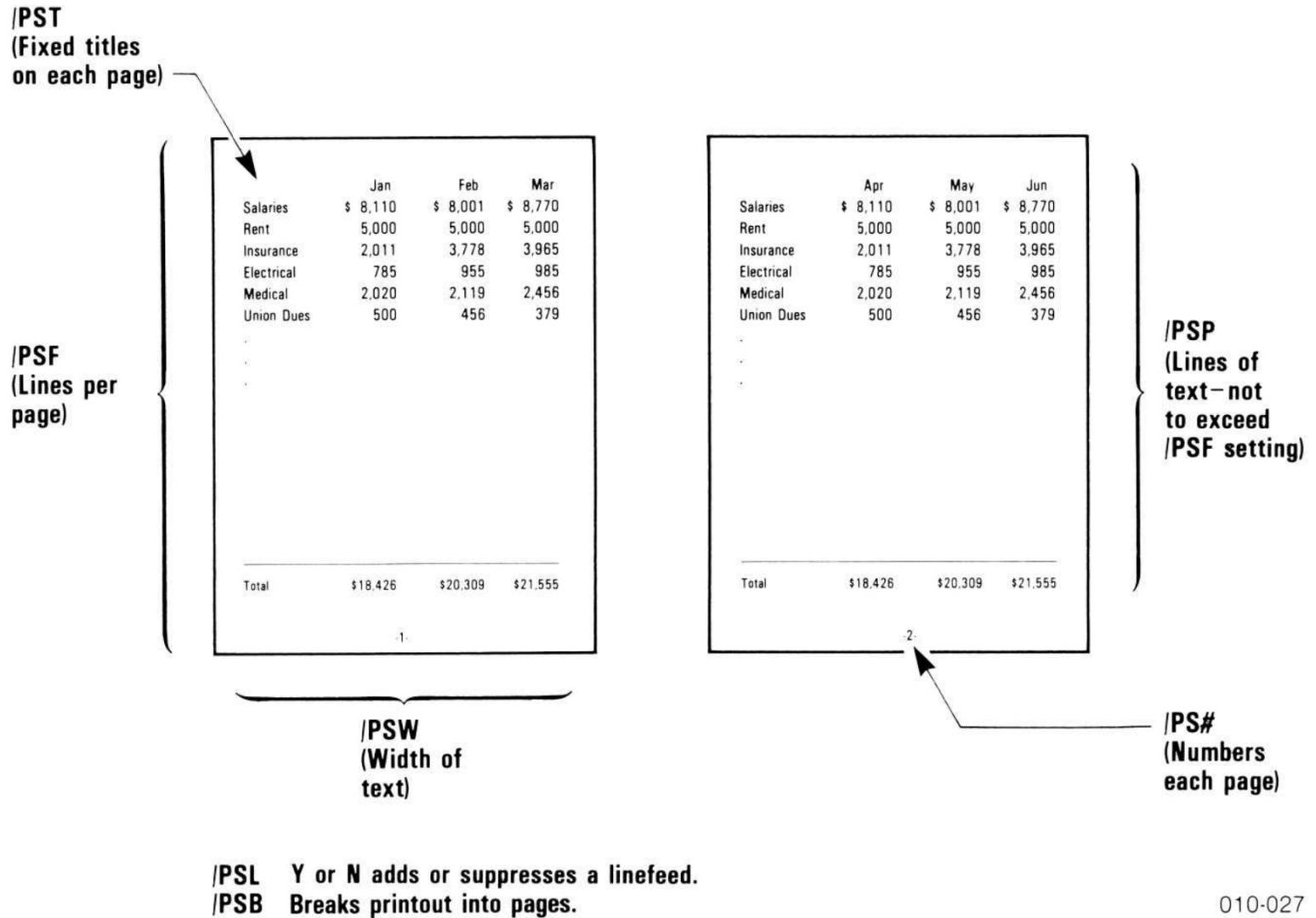
<b>Type:</b>	<b>Result:</b>
1. >A1 RETURN	The cursor moves to A1. This defines the upper-left corner of the rectangle to be printed.
2. /P	Prompt Line: Print: File, Printer, Settings
3. P	Prompt Line: Print: Lower right or "setup Prompt is asking you to either give the coordinate of the lower-right corner of the area you want printed or give some printer setup information. Edit Line: clear
4. J14 RETURN	The printer starts printing the worksheet.
5. CONTROL-C	The printer stops and the status area clears.

**Changing Printer-Setting Options—/PS**

The Printer-Setting options of the Print command let you control the appearance of printed worksheets. The following table shows the printer setting options along with a description of what each option does (default values are in parentheses).

<b>Setting</b>	<b>Default</b>	<b>What the option does</b>
L	(Y)	Inserts a blank line at the end of every printed line. /PSLN does not insert blank lines.
B	(Y)	Breaks worksheet into pages. /PSBN causes worksheet to print without page breaks.
T	(Y)	Prints column and row titles on each page of worksheet. Titles must be fixed by /T command before printing. /PSTN keeps titles from printing.
#	(Y)	Numbers each page of the worksheet. /PS#N causes pages to print out without page numbers.
P	(55)	Number of lines of text to be printed on each page; must be less than or equal to PAPER LENGTH option (/PSF). Can be from 0 to 255. To change, type an integer followed by RETURN or :.
W	(70)	Width in characters of each page (0-255). To change, type an integer followed by RETURN or :.
F	(66)	Length in lines of a page of printer paper on your printer (0-255). To change, type an integer followed by RETURN or :.
M	(0)	Left margin, in characters; offset from left edge of page to beginning of text (0-255). To change, type an integer followed by RETURN or :.
S		Edit setup string option. Setup string entered with /PSR appears on edit line; edit it with normal editing keys (ESCAPE, →, ←, etc.).
R		Replace or enter setup string. Type characters in setup string for your printer; end with RETURN.

The settings affect printed worksheets as shown in Figure 3-21.



**Figure 3-21. Printer-Setting Options**

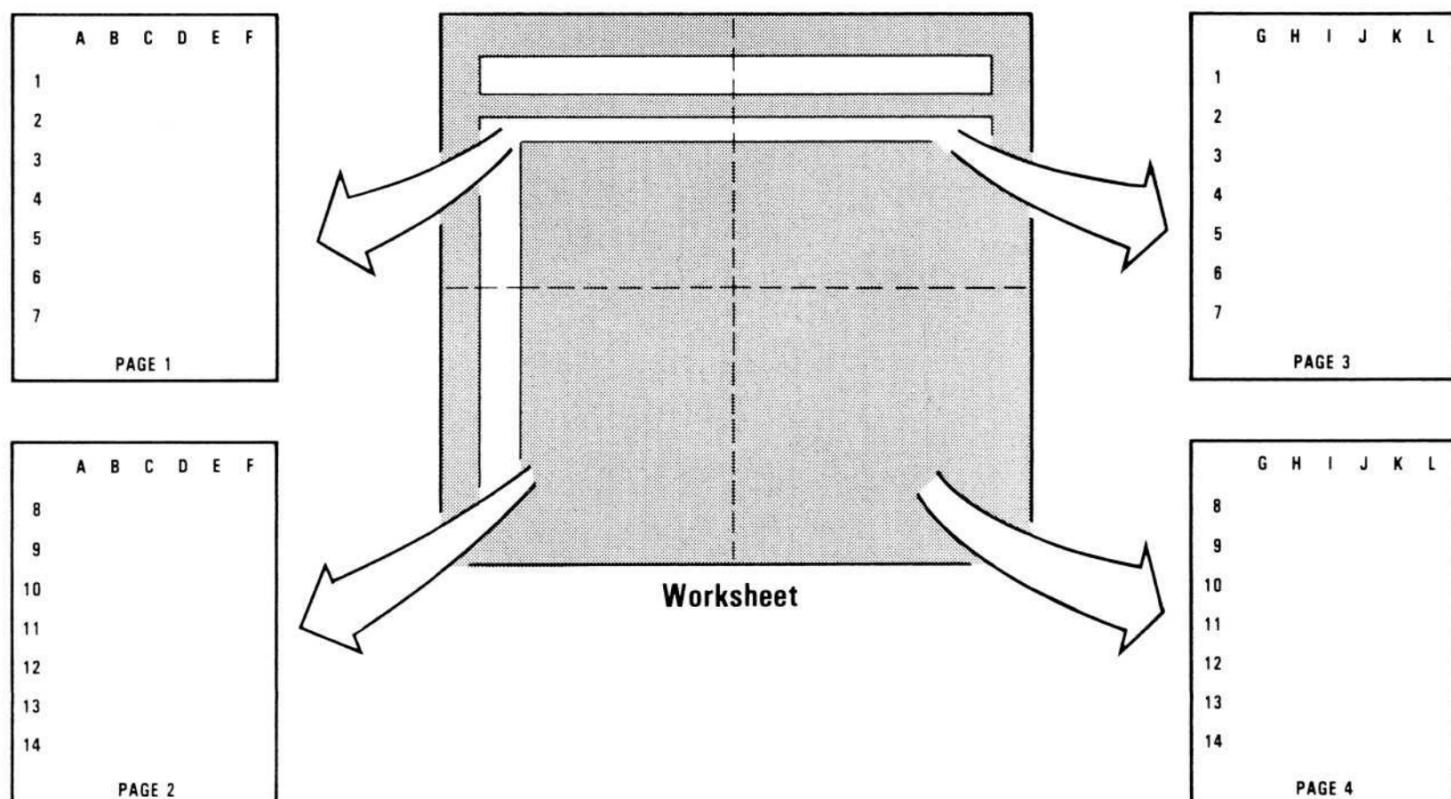
To review a specific printer-setting option, type /PS followed by the option. The current setting of an option follows its prompt. If you press RETURN without changing an option, the default is still in effect. To change an option, enter the appropriate change (depending on the option you are changing) on the Edit Line and, press RETURN.

To review all options, type /PS followed by →. Step through the options by pressing RETURN or →. To change an option, type /PS and the specific setting you want to change. Then type the option setting you want, followed by RETURN or →. You can then review the remaining options. The last option is /PSR, the replace printer setup string option. To enter setup string, type /PSR and the setup string; end with RETURN.

Printer settings in effect when a worksheet is saved are saved along with the worksheet. When a worksheet is loaded, the printer settings are also loaded; any current settings are overwritten.

**PAGINATION**

Worksheets are printed by column first, then by row, as Figure 3-22 shows.



010-028

**Figure 3-22. Printer-Setting Pagination**

**NOTE:** If the page-break option (/PSB) is set to N (no pagebreak), the Titles (/PST), page-number (/PS#), page length (/PSP) and paper length (/PSF) option settings are ignored.

**SENDING SETUP STRINGS**

A printer setup string is a series of instructions to your printer. The /PSS and /PSR Printer settings let you send a setup string to your printer each time you print a worksheet. After you type /PSR, you enter the setup string appropriate for your printer, followed by RETURN or : (see Appendix B for more information on printer setup strings).

Setup strings set with /PSR are saved along with the worksheet where they were typed. If you want to use the same setup string on several worksheets, you can type the setup string while a blank worksheet is displayed. One at a time, load each worksheet. Then load the blank worksheet with your setup string and save both together. Clear the screen with /CSY before loading the next worksheet.

To change a setup string entered with the /PSR option, type /PSS. The setup string will appear on the edit line, with the edit cue highlighting the first character in the setup string. Use the normal VisiCalc edit keys (ESCAPE, →, ←, ↓, ↑, etc.) to edit the setup string. When you have finished editing the setup string, press RETURN. The new setup string will take effect; it will be saved with worksheet if you use the save command (/S).

Setup strings sent with the /PP" option of the Print command are not saved in memory. If you type a setup string with /PSR and save it with the worksheet using /SS, then type another setup string when printing using /PP", the setup string sent with /PP" will be added to the setup string defined with /PSR. The /PSR setup string will still be in memory, however, and would still be saved with the worksheet.

## **PRINTING THE WORKSHEET IN SECTIONS**

If all /PS printer-setting options are set to N, and if the rectangle to be printed is wider than the number of characters the printer can print on one line, the part of the line beyond the printer limit is printed on a separate line. For example, if your printer prints 80 characters per line, the 81st character starts the next row down.

To reproduce a worksheet wider than the printer can print, print it in sections. Calculate how many columns of the worksheet can be printed at a time; then, divide the worksheet into the number of rectangles required to print the full width. The separate printed copies can be taped together to reproduce the worksheet.

If the printer uses single sheets rather than continuous-form paper, the worksheet might have to be printed in sections to accommodate the length.

You can also use the Printer Settings option of the Print command (/PS) to help you print your worksheet in sections. See the preceding section on "The Printer-Setting Options - /PS."

## Creating a Print File—/PF

The File option of the Print command (/PF) sends the image of the worksheet to a diskette file. A print file is a standard text file that contains exactly what is required to print the worksheet, including carriage returns and line feeds. It can be read by other programs, such as a word processor.

You might find it useful to add the suffix .PRF to the file name you specify. .PRF identifies the file as a print file, rather than a worksheet file, which should be given the suffix .VC or a DIF file which should be given the suffix .DIF. Because the file contains only the titles and numbers displayed on the worksheet (not the formulas, formats, etc., used to set up the worksheet), this file cannot be loaded with the Load option of the Storage command.

If you specify the name of a file that exists on the diskette, the VisiCalc program prompts you to make sure you want to replace the file: "File exists: Y to replace." If you type Y, the print file of the worksheet replaces the file with the same name on the diskette; if you type anything else, the Print command is canceled.

## DISKETTE FILES

Specifying a diskette file saves the image of the worksheet on diskette. See "File Names" in this chapter for a description of file names and how to specify them.

The following example shows how to save the area of the worksheet bounded by B10 at the upper-left and F20 at the lower-right in a print file named BUDGET.PRF on drive 1.

Example:

Type:	Result:
1. > B10	The cursor moves to B10.
2. /P	Prompt line: Print: File, Print, Settings The upper-left corner of the rectangle to be saved is defined as B10.
3. F	Prompt Line: Print: File name
4. budget.PRF RETURN	Prompt Line: Print: Lower right or "setup
5. F20 RETURN	Edit Line: F20 The worksheet is saved as a print file.

## **Printing Worksheet Formulas**

The Printer option of the Print command (/PP) prints an image of the worksheet as it is displayed on the screen. Normally, the screen display is numerical. But how do you print the formulas behind the numbers?

To print worksheet formulas, follow these steps:

1. Type /GAEY to put the entire worksheet in global expression display format.
2. If any cells were set locally to non-expression format (with /AEN), move the cursor to those cells and reset them to the global default with /AED.
3. Reset your column width, if necessary, to display every character in every formula on the worksheet. Move the cursor around the worksheet and compare the Entry Line to the cell display. If the cell display matches the Entry Line for each cell, your columns are wide enough. If not, make them wider with /GC or /GCC.
4. Issue the normal print command for your printer.

If you print worksheet formulas often, you might want to put a label in cell A1, or somewhere else on the worksheet, that tells what column width to set before printing. That way, you don't have to figure it out every time you print the worksheet.

## **Printing Formats, Attributes, and Formulas**

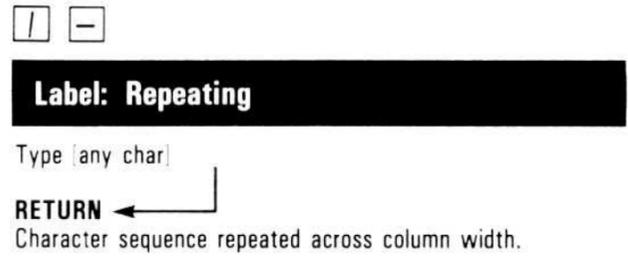
The Save option of the Storage command (/SS) saves a record of all formulas, formats and attributes as they were written on the worksheet. Like the other options of the Storage command, this record can be sent to a diskette file or printer. The information is so complete the worksheet can be reproduced from it.

To print a listing of each cell, including any formats, attributes, and formulas, follow these steps:

1. Type /SS.
2. Type .Printer.
3. Type RETURN.

The worksheet is printed cell-by-cell, from the lower-right cell to the upper-left cell, including format and attribute information. Each entry position takes one line, so a lot of paper is required to print a worksheet.

REPEATING LABEL COMMAND — / —



The Repeating Label command fills a cell with characters. It is commonly used to draw a line across the worksheet to separate titles from columns of numbers or to separate areas of the worksheet. The cell is filled with the character regardless of the column width.

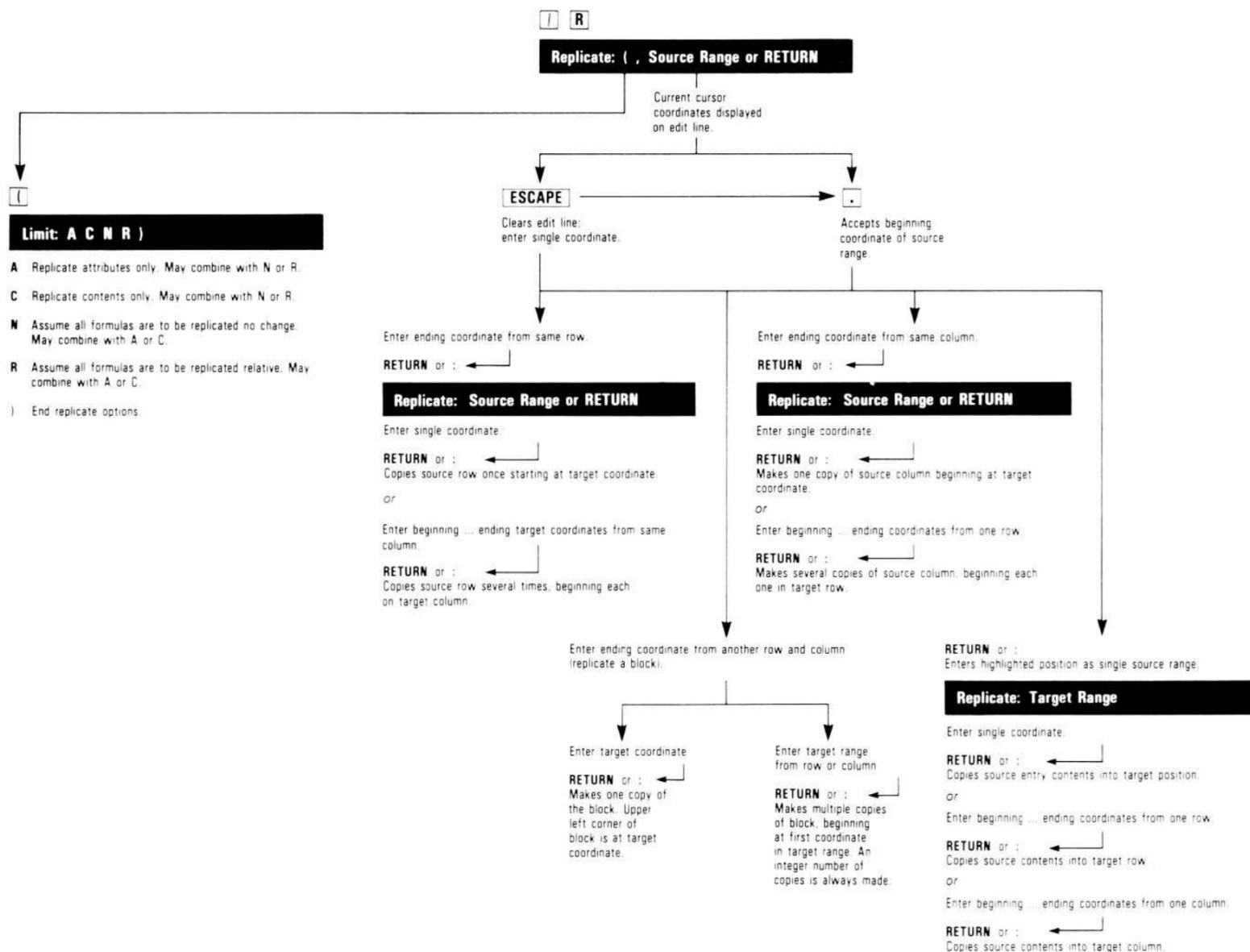
A column filled with characters can be copied across a row (with the Replicate command) to draw a line of any length across the worksheet.

Example:

<b>Type:</b>	<b>Result:</b>
1. /CAY	The screen clears.
2. Jan → Feb → Mar RETURN > A3 RETURN 1 → 2 → 3 RETURN	The numbers are separated from the titles by a blank line.
3. > A2 RETURN / —	Entry Line: A2 Prompt Line: Label: Repeating The program prompts for the character that is to fill the cell.
4. — RETURN	Entry Line: A2 /F- (L) — A2 fills with hyphens.
5. /GC20 RETURN	A2 is still filled with hyphens, even though the column is wider.
6. /GC9 RETURN	The columns return to the normal width.
7. /R RETURN → . → → RETURN	The line of hyphens extends across the first three columns.



## REPLICATE COMMAND—/R



The Replicate command copies a value, label, formula, blank entry, or the local format. It can create one copy of a single cell, multiple copies of a single cell, one copy of a range of cells, or multiple copies of a range of cells. It can also create one copy of a block of cells or multiple copies of a block.

The Replicate command cannot copy a row into a column or a column into a row. (See the section on "The Save Command" for information on how to do this.)

The Replicate command asks for the following information:

- The source range (cell or cells to be copied).
- The target range (cell or cells where the source range is to be copied).
- If the source range includes formulas that reference other cells, it asks whether each such reference should be copied with no change at each new location or relative to each new location.

A source range can be:

- One cell (for example, B5)
- A series of contiguous cells across a row or down a column (for example, F9-F12 or B2-M2)
- A block of cells in several rows and columns (for example, A1...C10).

The range (B2...M2) is defined by the starting coordinate (B2), an elipsis invoked by typing a period (...), and the ending coordinate (M2). If the range is a single location, the starting and ending coordinates are the same (for example, B2...B2). The coordinates can be specified by typing them or pointing to them with the cursor.

If the beginning and ending source-range coordinates are identical (for example, B5...B5), only the specified cell is copied to the target range. If the letters are the same but the numbers are different (for example, B5...B15), a portion of a column is copied. If the numbers are the same but the letters are different (for example, B9...M9), a portion of a row is copied. If both letters and numbers are different (for example, A5...D25), a block consisting of portions of rows and columns is copied.

To make one copy of a source range, the target range should consist of a single coordinate (for example, F9). To make more than one copy, the target range should consist of a range of locations (for example, F9...F15).

A colon separates the source range from the target range and is entered by typing either colon or RETURN.

	<b>source coordinate</b>	<b>:</b>	<b>target coordinate</b>
To copy one entry once			
Example:	A1...A1:D1		
To copy one entry several times	coordinate	:	range in one row or column
Example:	A1...A1	:	D1...G1
To copy a row or column once	row or column range	:	starting coordinate
Examples:	A1...G1	:	A10
	A1...A9	:	B1
To copy a row several times	row range	:	range in one column
Example:	D10...L10	:	D11...D15
To copy a column several times	column range	:	range in one row
Example:	B1...B9	:	C1...J1
To make one copy of a block	column and row range	:	starting coordinate
Example:	A1...B5	:	D1
To make several copies of a block	column and row range	:	range in one column or row
Examples:	A1...C5	:	A10...F10
	A1...C5	:	A10...A15

You can replicate cell contents only or cell attributes only by specifying an option of the Replicate command. See the following section on "Replicate Command Options" for details.

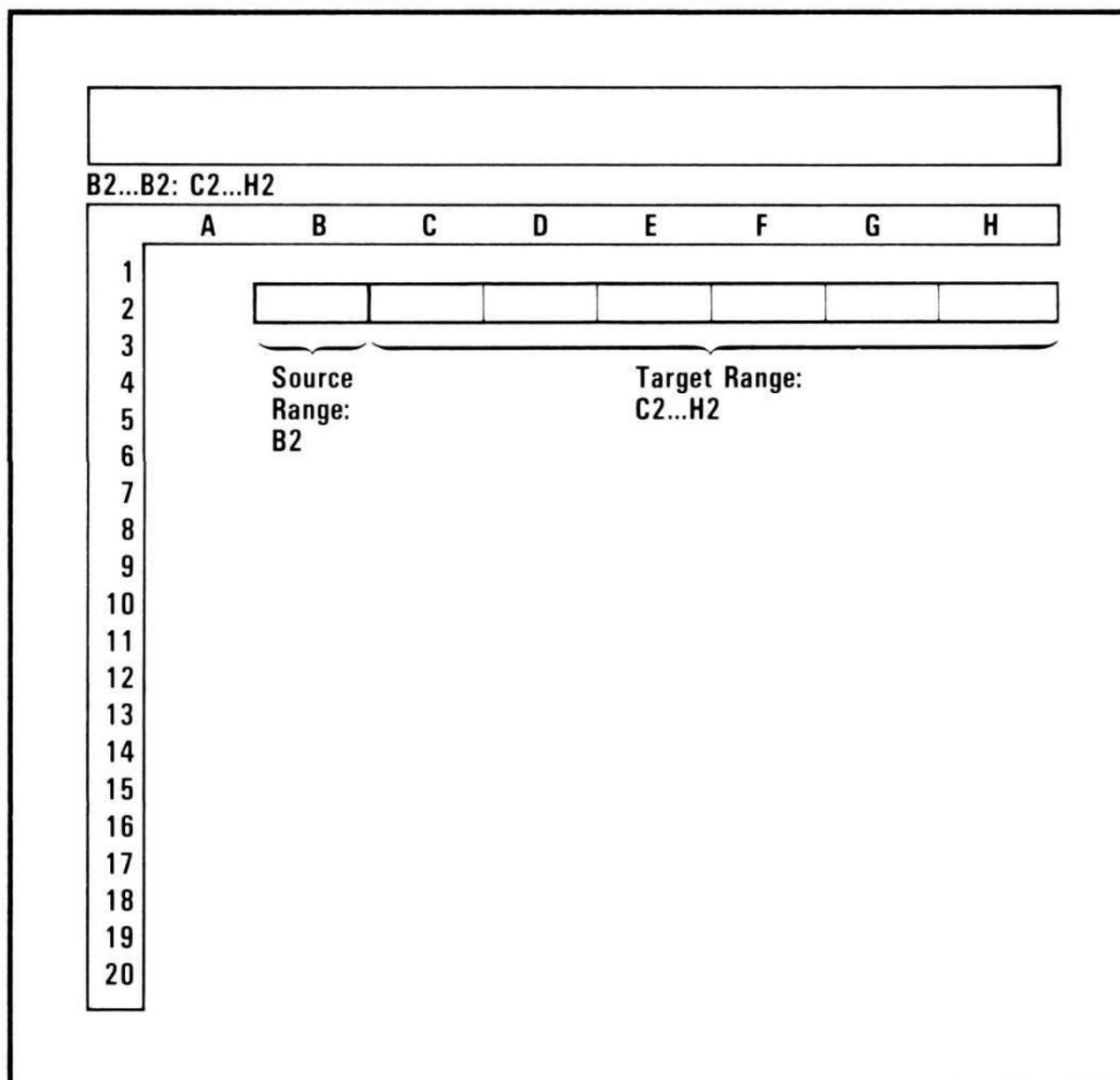
## Copying a Single Cell

To copy one cell, both the source and target range coordinates specify a single location.

Example:

Type:	Result:
1. <b>/CAY</b>	The worksheet clears and the cursor moves to A1.
2. <b>100 RETURN</b>	Entry Line: A1 (V) 100
3. <b>/R</b>	Prompt Line: Replicate: { Source range or RETURN Edit Line: A1 A1 is assumed to be the source range because the cursor is at that cell when the Replicate command is typed.
4. <b>RETURN</b>	Prompt Line: Replicate: Target range Edit Line: A1...A1 The source range is the single location A1.
5. <b>D1</b>	Prompt Line: Replicate: Target range Edit Line: A1...A1: D1 D1 is the target range.
6. <b>RETURN</b>	The value at A1 is copied to D1.

You can also replicate one cell several times, as shown in Figure 3-24, but the target range is then a row range or column range and is not a single cell.



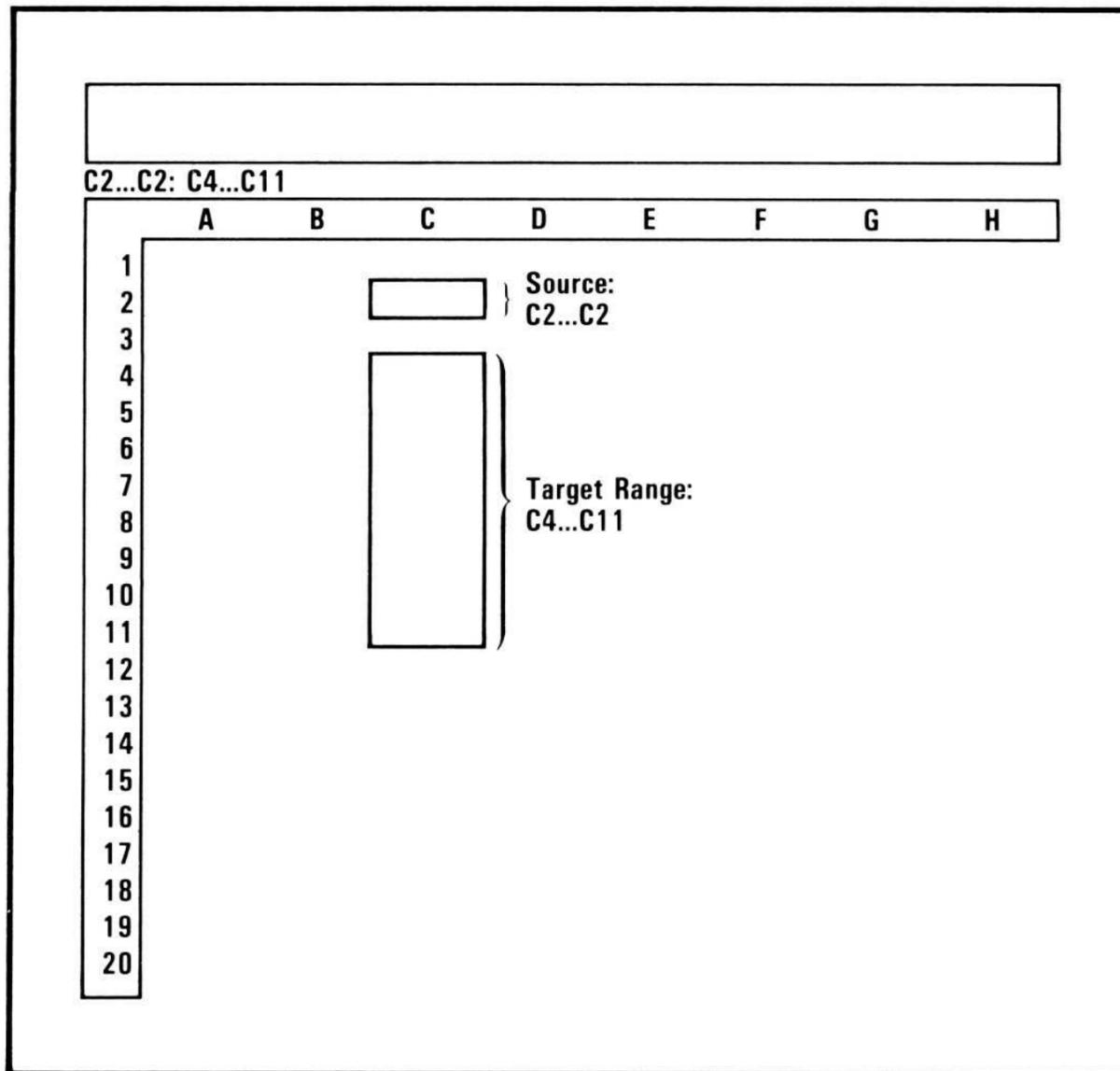
010-029

Figure 3-24. Replicating One Cell Across a Row

### Creating a Column of Entries from One Entry

To create a column by making several copies of one entry, specify the source range as the entry to be copied and specify the target range as the starting and ending coordinates of the portion of the column to receive the copy.

Figure 3-25 illustrates how to replicate an entry down a column.



010-030

Figure 3-25. Replicating One Cell Down a Column

## Making One Copy of a Column

To make one copy of a column, the source range is specified as the top and bottom locations (for example, A1...A32). The target range is specified as the coordinate of the top cell of the new column.

The cursor need not be located at the beginning of the source range. You can erase the starting source-range coordinate from the edit line with ESCAPE and type both the source and target ranges.

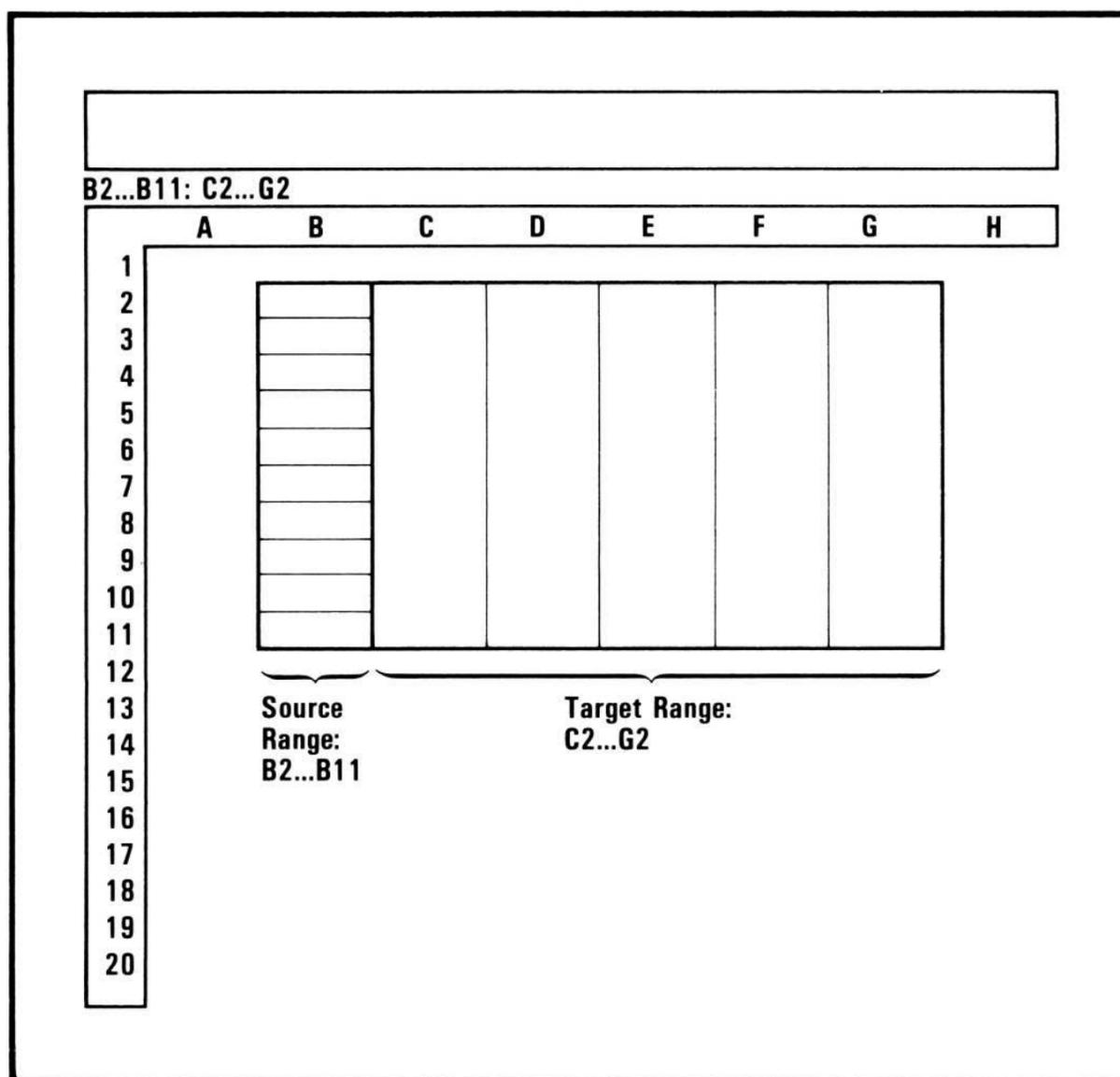
Example:

Type:	Result:
1. /CAY	The worksheet clears and the cursor moves to A1.
2. 1 ↓ 2 ↓ 3 ↓ 4 RETURN	A column of numbers in column A.
3. /R	Prompt Line: Replicate: [, Source range or RETURN Edit Line: A4
4. ESCAPE	Erases the first source coordinate.
5. A1	Edit Line: A1
7. A4 RETURN	Prompt Line: Replicate: Target range Edit Line: A1...A4:
8. C4	Edit Line: A1...A4: C4
9. RETURN	Column A (A1-A4) is copied to column C (C4-C7).

### Making Several Copies of a Column

To make several copies of a column, specify the top and bottom coordinates of the source range, and specify the target range as a range of cells across a row. The column is copied starting at each cell in the target range.

Figure 3-26 illustrates how to replicate a column several times.

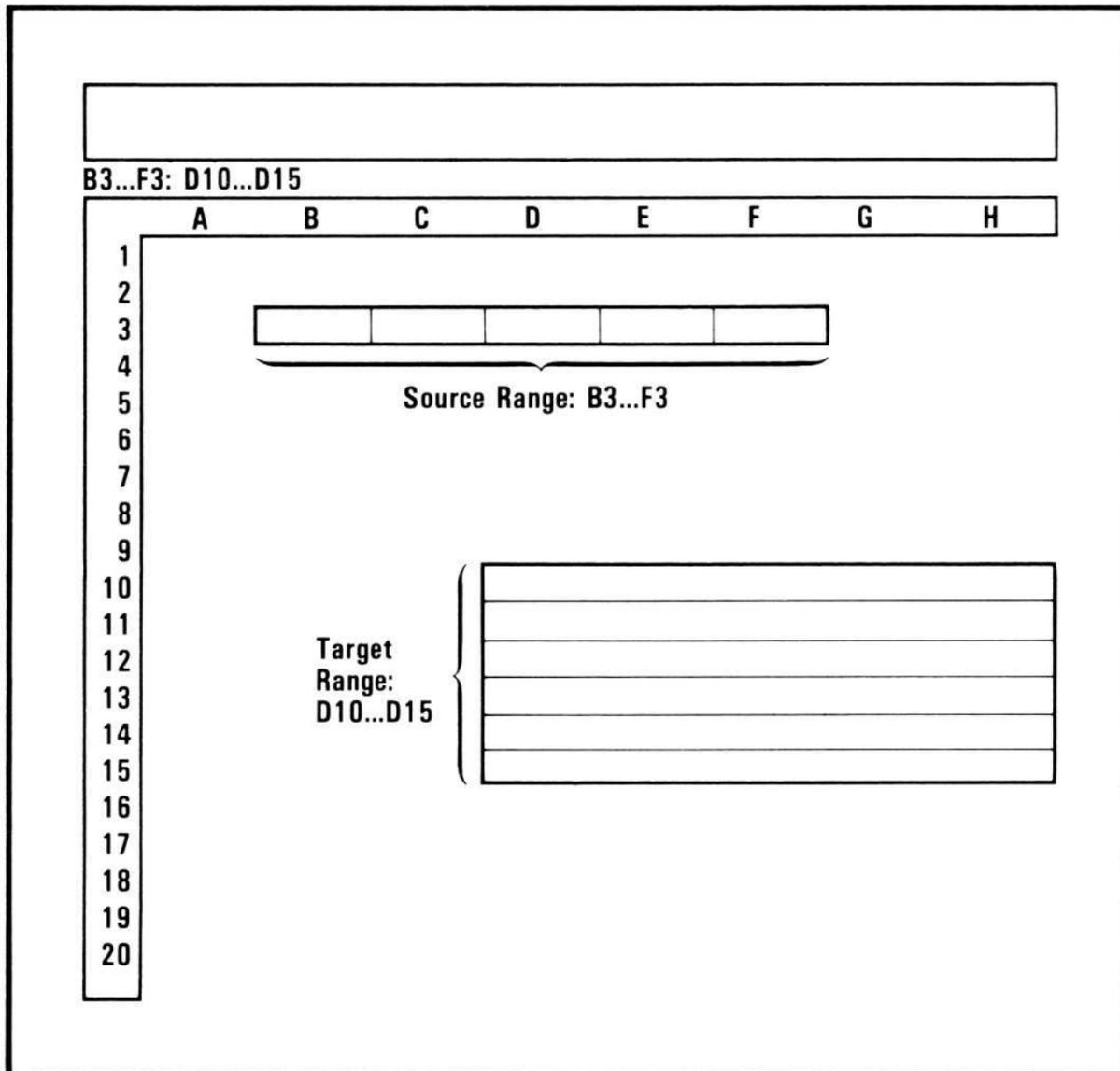


010-031

**Figure 3-26. Replicating a Column Several Times**

### Making Several Copies of a Row

To make several copies of a row, specify the beginning and ending coordinates of the source range row, and specify the target range as a range of cells down a column. The row is copied starting with each cell in the target range.



010-032

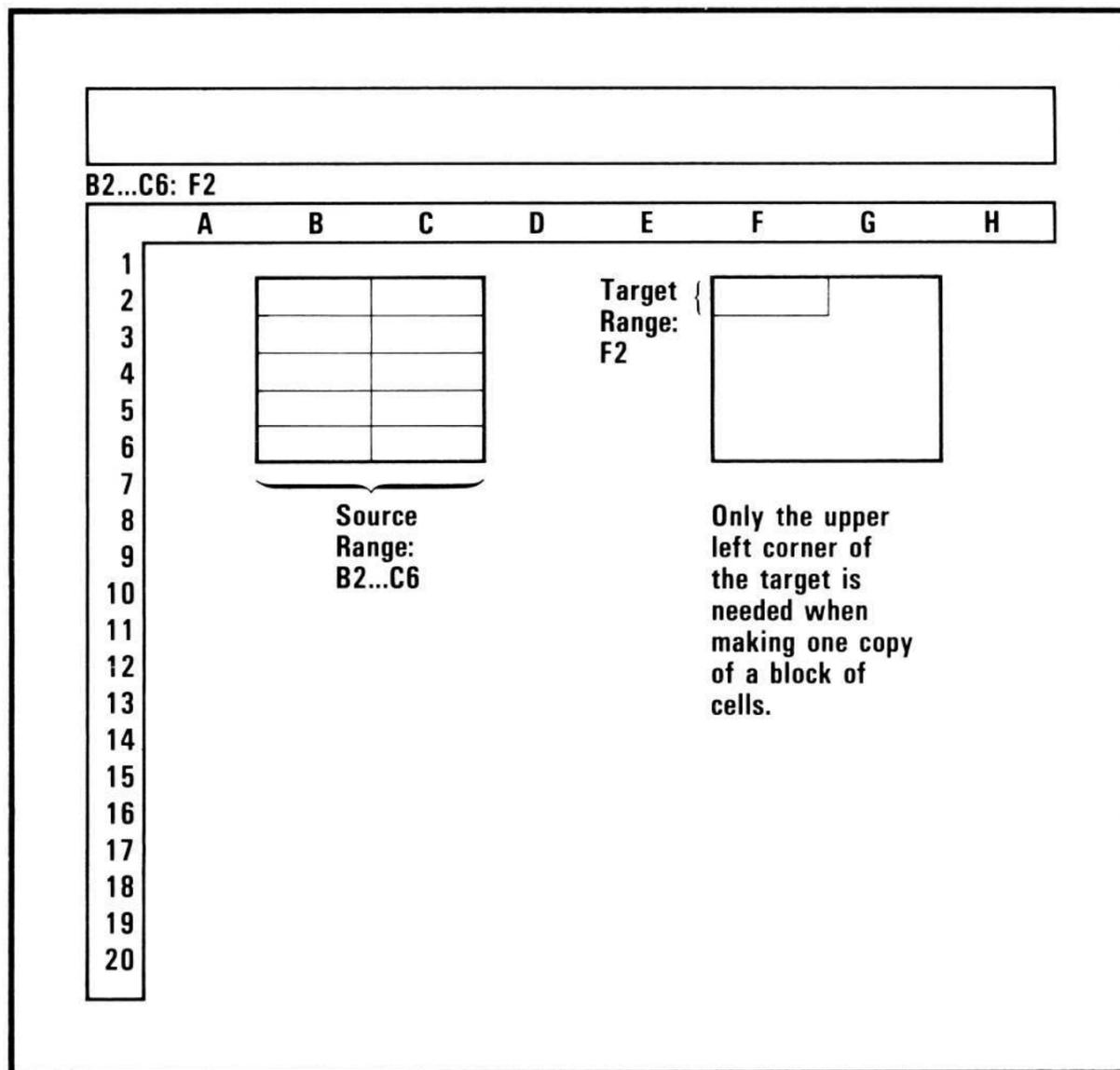
Figure 3-27. Replicating a Row Several Times

### Replicating a Block Once

A block is a rectangle of cells identified by the upper-left and lower-right coordinates, for example, A1...B2, D5...G30, and B2...C254.

To make one copy of a block, specify the source range as the upper-left and lower-right coordinates of the area to be copied (for example, A1...C5). Specify the target as the coordinate of the upper-left corner of the area to receive the copy.

Figure 3-28 illustrates how to make one copy of a block of cells.



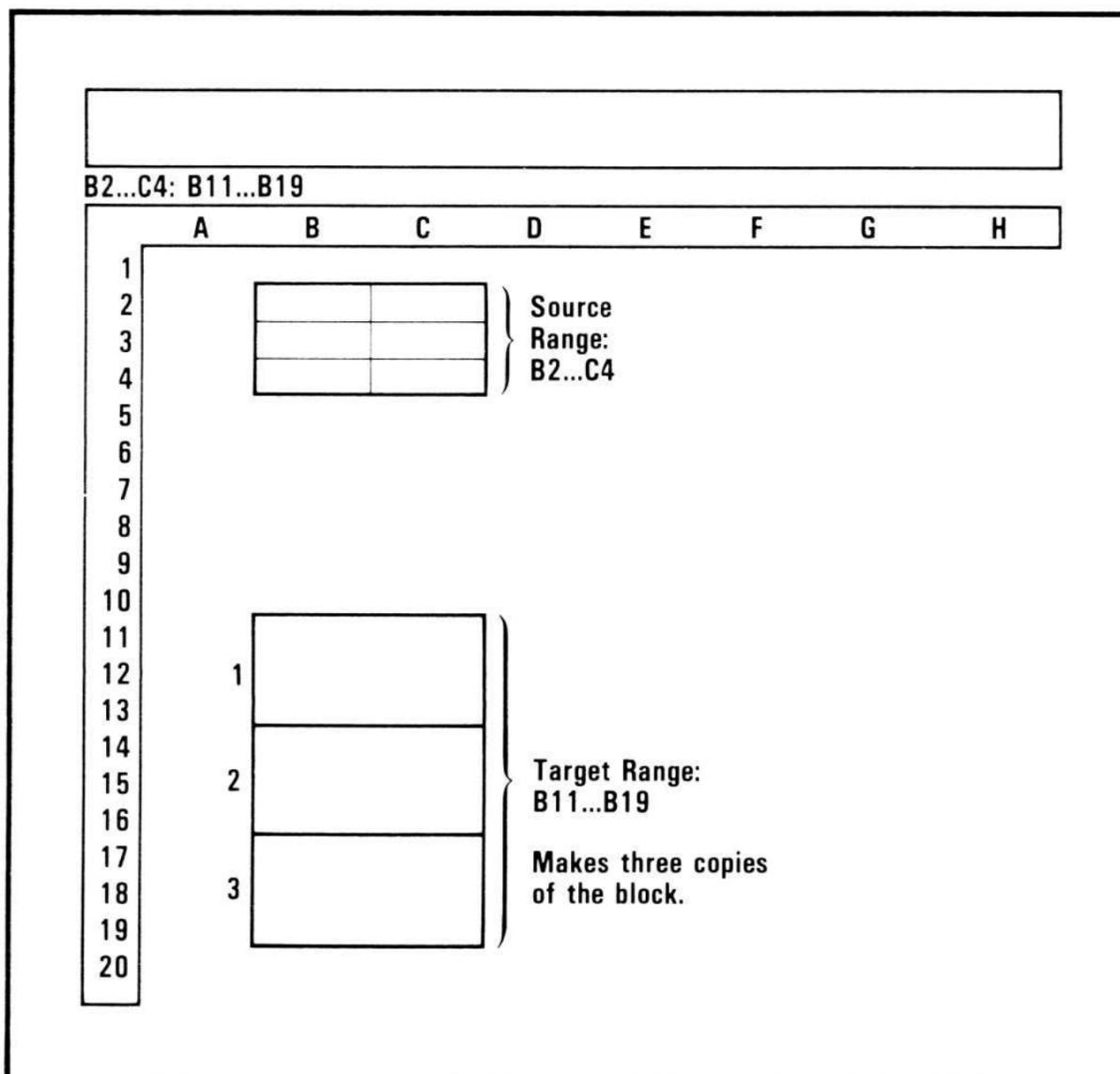
010-033

Figure 3-28. Replicating a Block Once

### Replicating a Block Several Times

To make several copies of a block, specify the source range as the upper-left and lower-right coordinates of the area to be copied (for example, A1...C5). Specify the target range as a range within one row or column, not as a block range. The last copy of the block will end in the row or column given as the second coordinate in the target range. An integer number of copies is always made, however. For instance, if you give A1...A5 as the target range when replicating a two-row block, three copies of the block will be made; the three copies are entered into rows 1 and 2, rows 3 and 4, and rows 5 and 6. Replicating blocks into columns follows the same rule.

Figure 3-29 shows how to make multiple copies of a block.



010-034

Figure 3-29. Replicating a Block Several Times

## Replicating Formulas

When the Replicate command copies a formula that contains one or more references to another cell (for example,  $.6 * B5 - A21$ ), it either copies the references with No change (N) or Relative (R) to the new location. The program prompts for an N or R for each reference in the formula being replicated.

If you want all cells to be replicated with No change, you can type (N RETURN after typing /R. If you do so, the program will not prompt for each reference in the formula. The formula will be replicated as if you had responded to all the prompts by typing N.

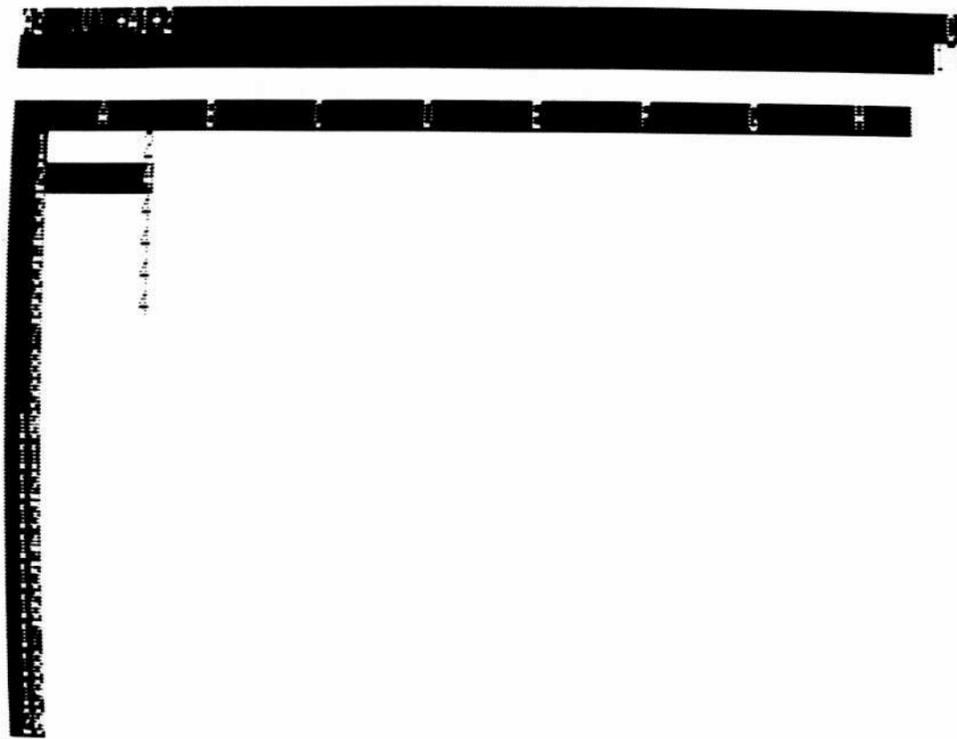
If you want all cells to be replicated Relative to their new location, you can type (R RETURN after typing /R. If you do so, the program will not prompt for each reference in the formula. The formula will be replicated as if you had responded to all the prompts by typing R.

Example:

Type:	Result:
1. /CAY.	The worksheet clears and the cursor moves to A1.
2. 2 ↓ +A1+2 RETURN /R RETURN	Entry Line: A2 (V)+A1+2 Prompt Line: Replicate: Target range Edit Line: A2...A2:  The RETURN after /R tells the VisiCalc program that the source range is a single location (A2).
3. A3.A6 RETURN	Prompt Line: Replicate: N=No change, R=Relative Edit Line: A2: A3...A6: +A1  The edit cue over +A1 indicates that the program is asking whether to copy the reference to location +A1 unchanged (the same in each position in the target range) or whether it is to be copied <i>relative</i> to each new position.

## 4. N

The reference to location A1 is copied with No change. The screen should look like Figure 3-30.



010-034/P

**Figure 3-30. Formula Replicated No Change**

Move the cursor down the column. Each copy of the formula contains the reference to A2, just as the source location does. This is the result of choosing No Change instead of Relative.

The next example chooses Relative and compares the results.

Example:

Type:	Result:
1. > B1 RETURN	The cursor moves to B1.
2. 2 ↓ + B1 + 2 RETURN /R RETURN	Entry Line: B2 (V) + B1 + 2 Edit Line: B2...B2 Prompt Line: Replicate: Target range The RETURN after /R tells the VisiCalc program that the source range is a single location (A2).

3. **B3..B6 RETURN**

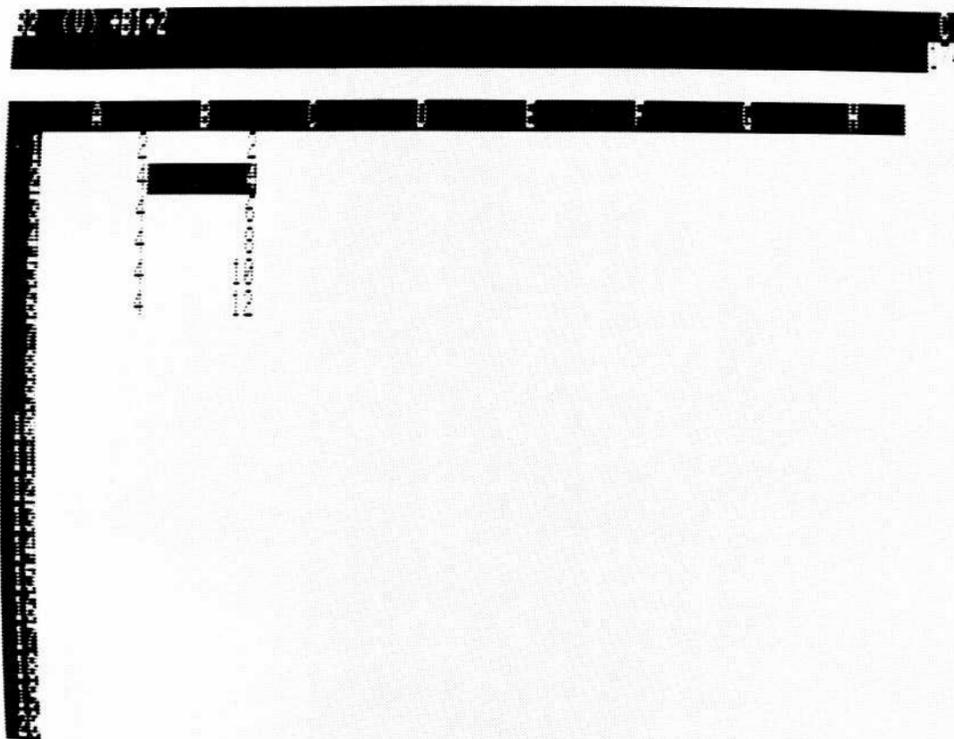
**Prompt Line:** Replicate: N=No change, R=Relative

**Edit Line:** B2: B3...B6: +B1

The edit cue highlights +B1. Again, it wants to know whether to copy the reference to location B1 unchanged (N) or relative (R) to the target locations.

4. **R**

The reference to location B1 is copied relative to each target location; that is, it becomes B2 in the formula at B3, B3 in the formula at B4, etc. The screen should look like Figure 3-31.



010-035/P

**Figure 3-31. Formula Replicated Relative**

The formulas behind columns A and B are quite different. Move the cursor down column B to see how the formula changes on each line. Now move down column A; each formula copied from A2 is the same.

If a formula contains more than one reference to another cell, the VisiCalc program asks how to copy each reference.

Example:

<b>Type:</b>	<b>Result:</b>
1. >C1 RETURN	The cursor moves to C1.
2. +B1/A1 RETURN	Entry Line: C1 [V] +B1/A1 The formula references two locations (B1 and A1).
3. /R RETURN	Edit Line: C1...C1: Prompt Line: Replicate: Target range
4. C2.C6 RETURN	Edit Line: C1: C2...C6: +B1 Prompt Line: Replicate: N=No change, R = Relative The edit cue is on the reference to B1, asking how to copy that reference.
5. R	Edit Line: C1: C2...C6: +B1/A1 Prompt line: Replicate: N=No change, R=Relative The edit cue is on the reference to A1, asking how to copy that reference.
6. N	The formula is replicated No change.

Move the cursor down column C to see how the dividend changes in each cell, but the divisor remains the same. To experiment further, replicate this formula with both dividend and divisor relative and with both No change. See "Replicating References to Other Cells" in Lesson Two for more examples of the Replicate command.

It is possible to replicate a formula to a position so that invalid relative value references are assigned. If you insert a new column at A and replicate the formula at C1 to A1, the relative position of the value references A1 and B1 are off the worksheet to the left. The VisiCalc program assigns the value ERROR to locations that contain such a reference.

Also be careful not to create forward or circular references or incorrect calculation order when you replicate formulas. See "Resolving Forward References" in Lesson Six, "Setting Manual Recalculation" in Lesson Seven and "Values" in this chapter for a description of these conditions and ways to avoid them.

## Options of the Replicate Command

You can specify one or two options of the Replicate command by typing /R(, one or two letters, and RETURN or :. Type ( to display the prompt Replicate: LIMIT: A C N R ). The options are described in the following list.

Option	What it Does
A	Replicates attributes or format only; cell contents are not replicated.
C	Replicates cell contents only; attributes are not replicated.
N	Replicates all cell references in a formula with No change.
R	Replicates all cell references in a formula Relative to their new position.
)	Ends Limit option; you can begin specifying source range.

You can select A or C, not both; and you can select N or R, not both. You can specify A, C, N, or R individually or in these combinations only: AN, AR, CN or CR. For instance, if you want to replicate all cell references in a formula with No change and you want to replicate only the contents of the cell (but not its attributes), type /R(NC RETURN or /R(CN RETURN before entering the source and target ranges. An option that conflicts with a previous option replaces the previous option; typing /R(NR is the same as typing /R(R, since the N is replaced.

### REPLICATING CELL REFERENCES IN FORMULAS: ALL RELATIVE OR ALL NO CHANGE-(R and N

The following example shows how to replicate all cell references in a formula with No change. This example uses the examples in the section on "Replicating Formulas."

Example:

Type:	Result:
1. >C1 RETURN	The cursor moves to C1.
2. +B1/A1 RETURN	Entry Line: C1 (V) +B1/A1 The formula references two locations (B1 and A1).
3. /R	Prompt Line: Replicate: [, source Range or RETURN

- |             |  |
|-------------|--|
| 4. (        | Prompt Line: Replicate: LIMIT: A C N R)        |
| 5. N RETURN | Prompt Line: Replicate: Source range or RETURN |
| 6. RETURN   | Edit Line: [N]C1...C1:                         |
| 7. C2.C6    | Edit Line: [N]C1..C1: C2...C6                  |
| 8. RETURN   | Both references are replicated No change.      |

Repeat the example, typing R RETURN in step 5 to see how to replicate all cell references Relative to their destination.

### REPLICATING ATTRIBUTES OR CELL CONTENTS-(A, (C

The following example shows how to replicate both cell contents and attributes with /R (no options); how to replicate attributes only with /R(A; and how to replicate cell contents only with /R(C.

Example:

- | Type:                       | Result:   |
|-----------------------------|---|
| 1. /CAY                     | Worksheet clears; cursor goes to A1.  |
| 2. /AV < +<br>RETURN RETURN | Cell A1 is assigned Value attributes that display values with + or - sign and align values with the left gutter of the cell.                            |
| 3. 100 RETURN               | Entry Line: A1 /A [V] 100   |
| 4. /R                       | Prompt Line: Replicate: [, Source range or RETURN   |
| 5. (                        | Prompt Line: Replicate: LIMIT: A C N R)   |
| 6. A)                       | The A (Attribute) option specifies that only the attributes in A1 are to be replicated.   |
| 7. RETURN → RETURN          | The attributes from A1 are now assigned to B1 as well.  |
| 8. /R(                      | Prompt Line: Replicate :LIMIT: A C N R)   |
| 9. C)                       | The C (Content) option specifies that only the contents of the cell are to be replicated.   |
| 10. RETURN ↓ RETURN         | The value 100 appears in cell A2. If you move the cursor to A2, you will see that there is no /A on the Entry Line; the attributes were not replicated. |
| 11. ↑/R                     | Prompt Line: Replicate<br>[, Source range or RETURN   |

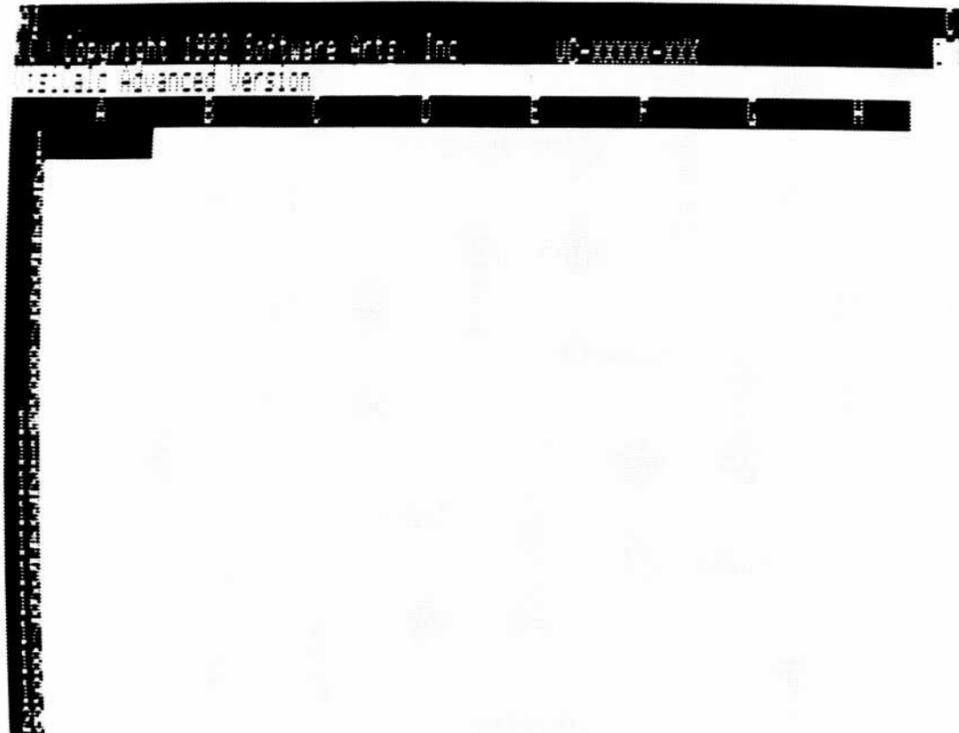
No option specifies that both the contents and the attributes in a cell are to be replicated.

**12. RETURN ↓ ↓ RETURN**

The value 100 appears in cell A3; the attributes in effect in cell A1 are now also in effect in cell A3 as well.

You cannot replicate some cell attributes and not others; all attributes in effect in a cell are replicated with /R(A).

**THE SCREEN**



010-036/P

**Figure 3-32. Startup Screen**

The screen includes the status area (the top three lines), the column and row borders, and the cells (locations) that make up the window through which you view the electronic worksheet.

## Status Area

The three lines of the status area are (from the top down) the *entry line*, the *prompt line*, and the *edit line*. The entry line and the prompt line share the wide white bar at the top; the edit line is the dark bar just below the white bar.

## Entry Line

The *entry line* is the top line of the status area. It can display as many as seven items of information about an entry position:

1. The coordinate of the cursor location. This is always displayed.
2. The local format, if one is assigned to the cell.
3. /A if there are any attributes assigned to the cell.
4. The type of entry, "L" for Label or "V" for Value, if the cell contains an entry.
5. The entry just as it was written, if the cell contains an entry.
6. A character that indicates the order of recalculation ("C" for column-order, "R" for row-order). An exclamation point is displayed next to it when the worksheet is being recalculated.
7. A "K" if a Keystroke Memory sequence is being defined with CONTROL-K = .

## Prompt Line

The *prompt line* is the middle line of the status area. It displays a prompt on the left that describes the options of a command sequence, and a number at the right side that tells how much memory is available (in Kilobyte units of 1024 characters).

## Edit Line

The *edit line* is the bottom line of the status area. It displays each character as it is typed, or the coordinate to which the cursor is pointing. The edit cue (a small white blinking block) appears after the last character entered.

The characters on the edit line can be edited by erasing them with the ESCAPE key and retyping the correct characters. The VisiCalc program also uses this line to confirm or clarify your information before it carries out a command.

## **The Window and Worksheet**

The VisiCalc *window* is that portion of the screen beneath the column border and to the right of the row border. The window can be scrolled with the arrow keys to show any portion of the worksheet.

The column border extends across the top of the worksheet and labels 63 columns, A through BK. The row border extends down the left side and labels rows 1 through 254.

## **Cells (Locations)**

The intersection of each column and row defines a *cell* or location. Each cell or location is identified by a column letter followed by a row number (A5, for example). This identifier is the coordinate of the cell or location.

## **The Cursor**

Each time the VisiCalc program is loaded or cleared with the Clear command, a column-wide white rectangle covers cell A1. This rectangle is the *cursor*.

The cursor marks the cell (location) on the worksheet where entries are written or commands are carried out. The coordinate of the cursor location is displayed at the left side of the entry line. You can move the cursor with the arrow keys.

## **Moving and Pointing the Cursor**

The arrow keys ( ← → ↑ ↓ ) move the cursor in the indicated direction to any position on the worksheet. Typing an arrow key moves the cursor one cell at a time. Holding down an arrow key moves the cursor in the direction of the arrow. Pressing harder while holding down an arrow key moves it faster. The cursor cannot move upward from row 1, downward from row 254, to the left of column A, or to the right of column BK. These are the edges of your worksheet.

When the cursor is at the edge of the screen, further cursor movement in the same direction causes the window to move over the worksheet, or *scroll*, until it reaches the edge of the worksheet. The Go To command moves the cursor directly to a location without scrolling.

If a command requires the coordinate of a cell, the coordinate can be entered in the formula by pointing to it with the cursor. As the cursor moves, the coordinate on the edit line changes. If a command does not require the coordinate of a cell, typing one of the arrow keys will end the command and move the cursor to the next entry position.

## Typeahead

If you type faster than the VisiCalc program can accept the characters, it stores the characters and catches up as soon as it can. If you type too fast, it cannot save any more characters and your computer makes a sound.

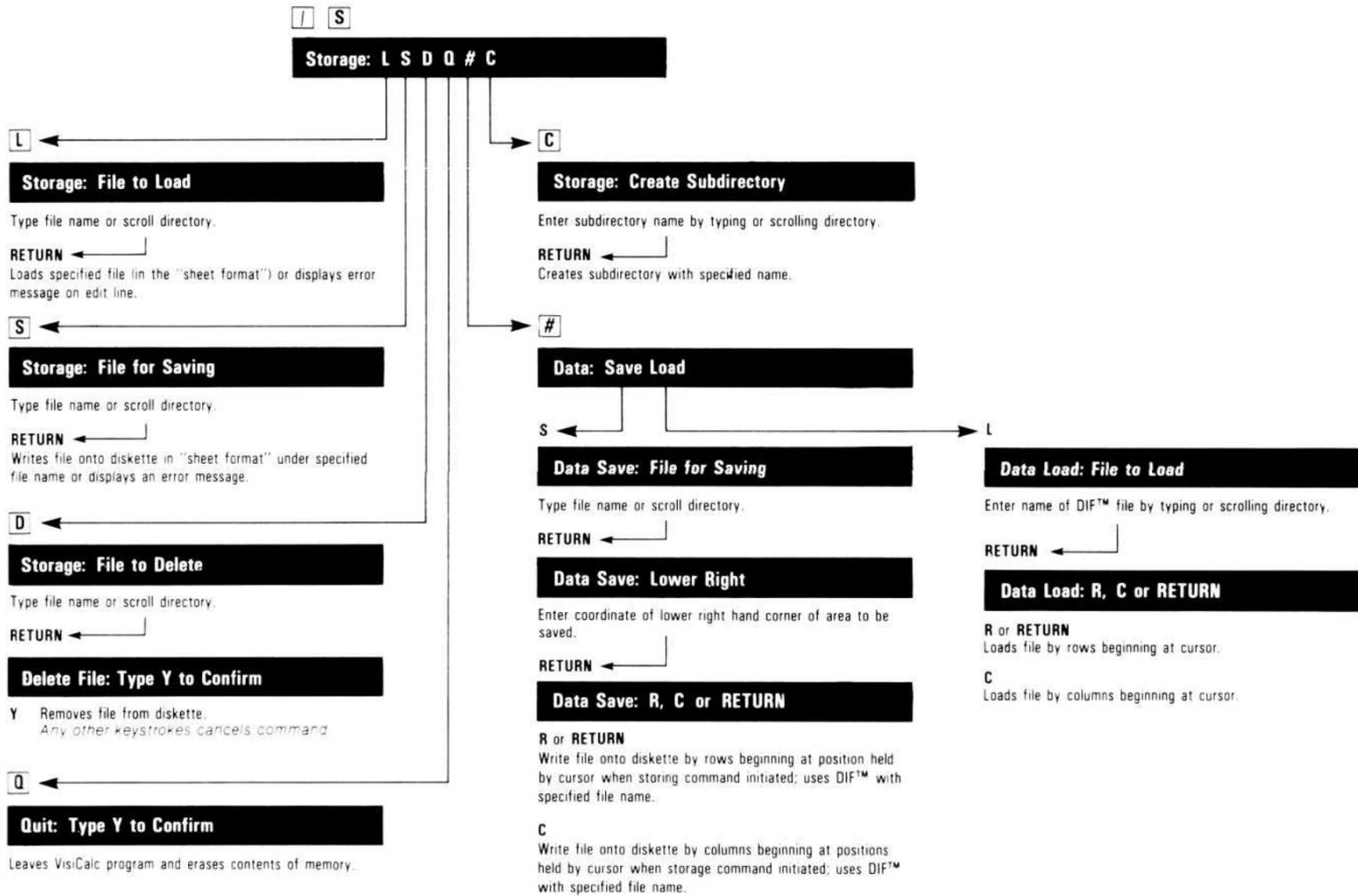
## Correcting Mistakes

You can correct mistakes several different ways (CONTROL-C, /B, typing over, CONTROL-E, and /E). CONTROL-C cancels a command or entry. The Blank command erases an entry from the worksheet.

An existing entry can be replaced by moving the cursor to the cell and typing a new entry. The new entry replaces the old one when it is accepted with RETURN or an arrow key. If the new entry is canceled with CONTROL-C before it is accepted with RETURN, the old entry is unchanged. The edit line shows the new entry as it is typed; the entry line shows the old entry until the new one is accepted.

The Edit command changes the contents of the edit line; it is used to correct an entry while it is being put on the worksheet (press CONTROL-E). You can use /E also to edit an entry after it is entered in a cell. With CONTROL-E, the edit cue is at the end of the Edit Line; with /E the edit cue is at the beginning of the Edit Line. The edit cue can be moved back and forth over the entry: → and ← move it one character right or left; ↑ moves it to the beginning of the edit line and ↓ moves it to the end. To erase the character to the left of the edit cue, press ESCAPE; to insert a character to the left of the edit cue, simply type the letter. To end the Edit command and write the entry on the worksheet, press RETURN.

## STORAGE COMMAND—/S



The Storage command has five options:

- L Load a previously saved worksheet.
- S Save the VisiCalc worksheet on diskette.
- D Delete a file from the storage diskette.
- Q Quit the VisiCalc program.
- # Load or save a file in DIF format.
- C Create a Subdirectory.

When you save the worksheet, the Storage command records the formulas, formats, attributes, printer settings, Keystroke Memory sequences and other instructions that produce the worksheet (not the image of the worksheet that appears on the screen, which is saved by the Print command). This record can be sent to:

- A printer.
- A diskette file.

**NOTE:** A diskette must be formatted before anything can be written on it. See your *Apple III Owner's Manual* for instructions on how to format a diskette.

## Load Option—/SL

The Load option loads a worksheet that was saved with the Save option. When a worksheet is loaded, it is displayed exactly as it was saved. If it contains forward or circular references, those cells display `ERROR`. (If `ERROR` appears while the file is loading, ignore it.)

If one worksheet is on the screen when a second one is loaded, the first does not clear and the second worksheet loads over the first. If a cell is filled on both worksheets, the entry on the second worksheet replaces the entry on the first. If a cell on the second worksheet is blank, the corresponding cell on the first is not erased. This makes it possible for one worksheet to be combined with another. If you do not want to combine the two worksheets, clear the screen (with the Clear command) before the second worksheet is loaded.

Keystroke Memory sequences defined for the first worksheet will be replaced by sequences with the same name defined on the second worksheet.

A sequence that loads a worksheet executes completely, even if the worksheet being loaded includes a sequence with the same name as the loading sequence.

The following example shows how to load a worksheet named "budget.VC" from drive 1:

Example:

Type:	Result:
1. /S	Prompt Line: Storage: L S D Q # C
2. L	Prompt Line: Storage: File to load
3. budget.VC RETURN	During the loading process, the prompt line reads Storage: Loading followed by a flashing asterisk. When the worksheet is loaded, it appears on the screen just as it did when it was saved (including any split windows, fixed titles, and global format).

Instead of typing in the file name, you can display the name of each file on the diskette by typing → and select the one you want by pressing RETURN. Type → instead of "budget.VC" at step 3 in the preceding example. The name of the first file on the diskette is displayed on the edit line. Type → again. Each time you type → the file name of each file on the diskette is displayed. (If you type → again, the Storage command is canceled.) When the file name you want is on the edit line, press RETURN, and the VisiCalc program loads the worksheet. After the first time you type →, typing ← displays the name of the preceding file.

These file names can be edited. To edit the file name on the edit line, type CONTROL-E or ESCAPE, and edit the line as described in "Edit Command" in this chapter. When the name is correct, press RETURN. This feature is useful when, for example, you have forgotten a file name. The name "projfeb", for example, might remind you that the file you want is called "projaug". To change the file name from "projfeb" to "projaug", press ESCAPE to back up the edit cue until you've deleted "feb"; then type aug RETURN. The worksheet saved in the file named "projaug" is loaded.

If the VisiCalc program cannot find the file you tell it to load, the prompt line reads ERROR: Not found. This could mean you misspelled the name, the wrong diskette is in the drive, or the wrong drive is being searched. Correct the name, put in the proper diskette, or add the proper drive number (preceded by a period (.)) and followed by a slash (/) to the beginning of the file name.

**NOTE:** If you hear many beeps when loading a file, the file is full of circular and/or forward references. If you see EOD in A1, you are trying to load a data (DIF) file with /SL instead of /S#L.

### **Save Option—/SS**

The Save option of the Storage command (/SS) saves a record (stored image) of the worksheet. The record can be sent to a file or a printer.

See "Print Command" in this chapter for a description of how to specify a printer. See "File Names" for a description of how to specify a file name.

If you specify the name of a file that is already on the diskette, the VisiCalc program prompts you to make sure you want to replace the existing file: `Storage: File exists. Y to replace.` If you type `Y`, the file you are saving replaces the file with the same name; if you type anything else, the `Storage` command is canceled.

The ability to display the names of the files on a diskette makes it easy to name files as new versions of existing files. For example, you could modify a file called `"sales5"` and, instead of replacing it with the new version, display the file name, press `ESCAPE` to erase the `5`, then type `6`. The result: a new version named `"sales6"`; `"sales5"` is still on the diskette.

Although a diskette has enough room to hold many worksheets, it still has a finite amount of space. If the disk fills while a worksheet is being saved, `Error: Disk full` appears on the edit line; the VisiCalc program saved only part of the worksheet. If this happens, delete the incomplete file from the disk (see `"Delete Option"` below) and save the worksheet on a diskette with enough room.

The following example saves the worksheet in a file named `"budget.VC"`.

Example:

Type:	Result:
1. <code>/S</code>	Prompt Line: <code>Storage: L S D Q # C</code>
2. <code>S</code>	Prompt Line: <code>Storage: File for saving</code>
3. <code>budget.VC RETURN</code>	The status area clears, the drive activates, and the worksheet in the file named <code>"budget"</code> is saved.

## Delete Option- /SD

The Delete option of the Storage command (/SD) *irretrievably erases* a file from the diskette.

The following example deletes a file named "budget" from the diskette in drive 1:

Example:

Type:	Result:
1. /SD	Prompt Line: Storage: File to Delete
2. budget	Edit Line: budget
or	
.D1/ →	Edit Line: The name of the first file on the diskette in drive 1. Each subsequent → causes the next file name to be displayed. Continue until the name of the file you want is displayed (if you type → after the last file name is displayed, the Storage command is canceled).
3. RETURN	Prompt Line: Delete file: Type Y to confirm.
4. Y	The file is deleted from the diskette. If you type anything other than Y, the Storage command is canceled.

## Quit Option- /SQ

The Quit option of the Storage command (/SQ) returns control to your computer's operating system. It first prompts *Quit: Type Y to confirm*. If you want to quit the program and start another program, type **Y**. If you don't want to start another program, simply remove the diskettes and turn the computer off.

## DIF™ Options-S#S and S#L

The DIF™ options—S#S and S#L—are used to save and load worksheets stored in the DIF format. With these options, you can exchange files with other programs. For a complete description of the DIF format data structure, see Appendix C.

## Save DIF™ Option-/S#S

The Save DIF option of the Storage command (/S#S) saves the worksheet in the DIF format. You might find it useful to add ".DIF" to the file name so it can be recognized as a worksheet saved as a DIF file.

Values are saved to their full precision. This precision might cause the saved worksheet to differ from the displayed version. To save the worksheet with all values calculated to the same precision as the version on the screen, use the @ROUND function (see "Using @ROUND to Round for Less Precision" in the "Functions" section of this chapter), or save the file with the File option of the Print command (/PF) and convert it to the DIF format with a separate program, e.g., in BASIC.

The VisiCalc program saves and loads a DIF-file worksheet by row or by column. The VisiCalc program prompts you to specify the row or column option by typing **R** or **C**. Specify **R** or **C** or press **RETURN** when saving a worksheet to be loaded later by the VisiCalc program.

The following example saves the area of the worksheet whose upper-left boundary is **A1** and lower-right boundary is **J14** in a file named **BUDGET.DIF** on the disk in drive 1:

Example:

<b>Type:</b>	<b>Result:</b>
1. <b>&gt;A1 RETURN</b>	The cursor moves to <b>A1</b> .
2. <b>/S#</b>	Prompt Line: Data: Save Load
3. <b>S</b>	Prompt Line: Data save: File for Saving
4. <b>budget.DIF RETURN</b>	Prompt Line: Data save: Lower right Edit Line clears.
5. <b>J14 RETURN</b>	Prompt Line: Data save: R C or RETURN Or you can move the cursor to <b>J14</b> .
6. <b>R RETURN</b>	The prompt line clears and the drive activates, saving the sheet as a <b>DIF</b> file.

## Load DIF™ Option—/S#L

The Load DIF option of the Storage command (/S#L) loads a worksheet stored on a diskette in the DIF format. The file can be one saved with the Save DIF option of the Storage command, or it can be created by another program.

A normal worksheet is loaded into the same position it occupied when it was saved. In contrast, a DIF worksheet is loaded starting at the position where the cursor is when the Storage command is invoked.

The following example loads the worksheet stored in a file named BUDGET.DIF on the diskette in drive 1 starting at cell C10:

Example:

Type:	Result:
1.. /CAY	The worksheet clears, and the cursor moves to A1.
2. >C10 RETURN	The cursor moves to C10.
3. /S#L	Prompt Line: Data load: File to Load
4. budget.DIF RETURN	Prompt Line: Data load: R C or RETURN
5. R	The worksheet appears. Cell C10 is in the upper-left corner.

You can display the names of the files on the diskette by typing → in response to the File to load: prompt.

To load across rows instead of down columns, type C instead of R at step 5. A worksheet saved in the column format can be loaded in row format by typing R at this point.

## Creating Subdirectories: The /SC Command

You might want to create a subdirectory to group together related files. A subdirectory is just like a file in the main directory, except that it contains one or more files or subsubdirectories.

A subdirectory is listed in the directory just like a file, except that its name ends with a slash. The files in the subdirectory are not listed in the directory—instead, they are listed in the subdirectory. Each subdirectory can have its own subsubdirectories, and their names are listed in the subdirectory just like files, except that they also end with slashes.

Insert your storage diskette in drive 2.

Example:

Type:	Result:
1. /SC.D2/	Prompt Line: Storage: Create subdirectory
2. →	Edit Line: .D2/
3. SUBDIR1 RETURN	The prompt and edit lines clear; your subdirectory named SUBDIR1 has been added to the directory.

To add a file to the subdirectory, first add information to a sheet so the file won't be blank (this makes it easier to verify that the procedure worked).

Example:

Type:	Result:
1. /CAY SUBDIR1 RETURN	Clears the worksheet, and puts SUBDIR1 at A1.
2. /SS →	Saves the file. Edit Line: .D2/SUBDIR1/ (If something else is on the edit line, press → until .D2/SUBDIR1/ appears.)
3. FILE1.VC RETURN	The drive whirs, and the prompt line clears.

To load the file just created:

Type:	Result:
1. /CAY	Clears the worksheet.
2. /SL →	Edit Line: .D2/SUBDIR1/ FILE1.VC
3. RETURN	The drive whirs, and the file loads. Position A1: SUBDIR1

To create a second subdirectory:

Type:	Result:
1. /CAY	Clears the worksheet.
2. SUBDIR2 RETURN	Position A1: SUBDIR2
3. /SC →	Edit Line: .D2/SUBDIR1/ When you scroll through the directory after entering /SC, only subdirectory names are displayed.
4. ESCAPE ESCAPE	Edit Line: .D2/SUBDIR
5. 2 RETURN	Creates a subdirectory with the name SUBDIR2.

To edit names on the edit line another way, at step 4 above, press ↑. When you are scrolling through a directory, pressing ↑ backs up the edit cue to the previous slash (in this case, the slash after .D2). This option is particularly useful if you want a completely different subdirectory name. You can also type **CONTROL-E** to edit more extensively.

Now save the worksheet by typing:

**/SS → FILE2.VC RETURN**

To retrieve a file from a subdirectory:

Type:	Result:
1. /SL →	Scroll with → until the edit line reads .D2/SUBDIR2/
2. Press RETURN	Edit Line: .D2/SUBDIR2/ FILE2.VC

Pressing RETURN when the edit line displays a subdirectory initiates scrolling through the subdirectory. If more files are in the subdirectory, you can get them by pressing →. If you want to leave the subdirectory, type ↑↑ to return to the prefix .D2/, and continue from there.

For the purposes of illustration, the examples above explored all the different ways of entering a file name on the edit line. Other ways exist:

1. You can just type the desired file name. In this case, the name must be exactly right; use scrolling if you can't remember the exact name.
2. You don't have to enter the default prefix. If the default prefix applies to the file name you want, type the local file name without the prefix. Again, if you can't remember the exact name, use scrolling.

The following is a summary of the effect of pressing various keys during the scrolling process:

→ Causes the next entry in the current directory to be displayed on the edit line. In the /SC command, it displays the next subdirectory name.

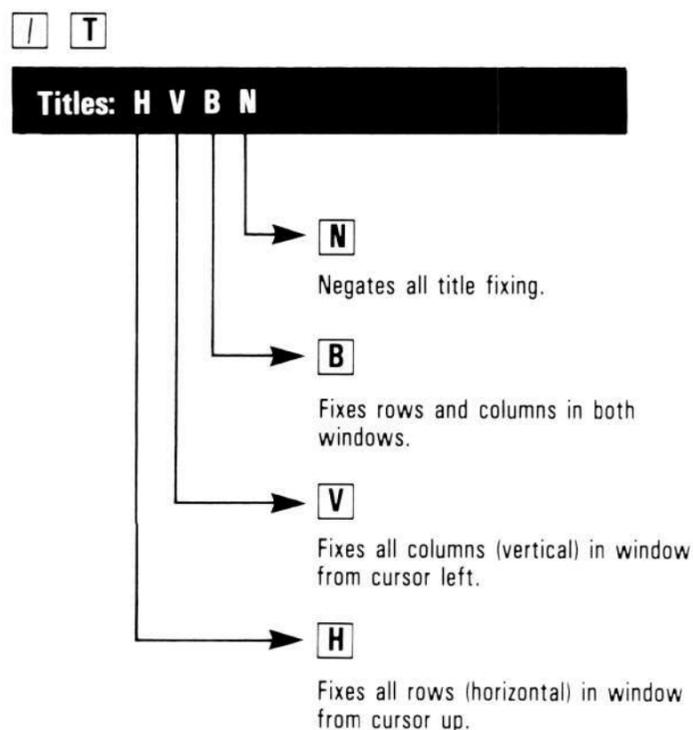
← Causes the previous entry in the current directory to be displayed on the edit line. In the /SC command, it displays the previous subdirectory name.

↑ Backs up the edit cue to the previous slash, deleting characters in between.

RETURN If the edit line contains a complete pathname, pressing RETURN accepts the name for the current command. If the edit line contains a subdirectory name, pressing RETURN enters that subdirectory.

ESCAPE Backs up the edit cue one character.

CONTROL-E Enters the Edit command to allow editing of the file name. Pressing RETURN accepts the name if it is a complete pathname, or enters the subdirectory if it is a subdirectory name.

**TITLES COMMAND- /T**

The Titles command fixes rows and columns to remain in view as the window scrolls over the worksheet. The position of the cursor at the time the Titles command is entered determines which column(s) and/or row(s) are fixed.

In response to `/T` (the Titles command), the prompt line displays Titles: H V B N. The characters following Titles: are options:

- H Fixes Horizontal titles.
- V Fixes vertical titles.
- B Fixes Both horizontal and vertical titles.
- N No titles (unfixes all titles on the worksheet).

The Titles command does not change the appearance of the worksheet. The effect of title fixing becomes apparent when the window scrolls; the columns and rows that are not fixed move across the screen, but the fixed titles do not move.

You cannot use the arrow keys to move the cursor into a fixed-title row or column. The cursor beeps when it bumps into a fixed title. However, you can use the Go To Command (`>`) to move to a cell in a fixed-title row or column.

If vertical titles are fixed and the column width is increased so that only one column can be displayed, the VisiCalc program cancels the fixed vertical title settings. If the column width is later reduced, vertical titles must be fixed again.

### **Horizontal Option—/TH**

The Horizontal option of the Titles command fixes all rows at and above the row that contains the cursor. To fix rows 1 and 2 as titles, place the cursor anywhere in row 2. To fix only row 1, place the cursor in row 1.

### **Vertical Option—/TV**

The Vertical option of the Titles command fixes all columns at and to the left of the column that contains the cursor. To fix columns A and B, place the cursor anywhere in column B. To fix only Column A, place the cursor in column A.

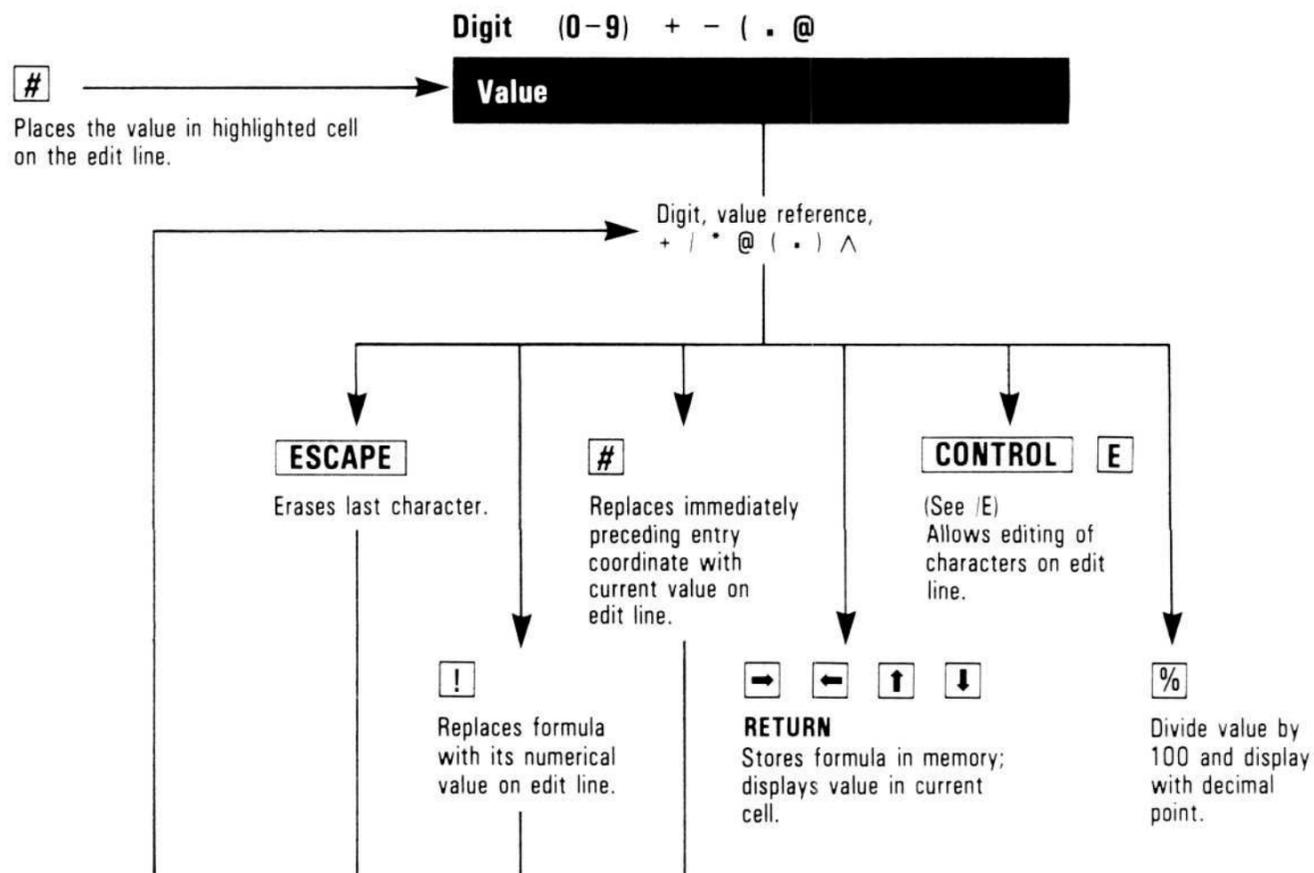
### **Both Option—/TB**

The Both option of the Titles command fixes Both rows and columns at the same time. The rows are fixed at and above the row that contains the cursor, and the columns are fixed at and to the left of the column that contains the cursor. To fix row 1 and column A, place the cursor at location A1. To fix rows 1 and 2 and columns A and B, place the cursor at location B2.

### **None Option—/TN**

The None option of the Titles command unfixes all fixed titles on the worksheet. The cursor can be anywhere when the command is entered.

## VALUES



A value is an entry on the worksheet used in calculations. The VisiCalc program evaluates the first character of the entry as a value or a label; if the entry begins with any of the following characters, it is a value:

- 0-9 The digits zero through nine.
- + Plus sign.
- Minus sign.
- @ The beginning of a function name.
- ( Left parentheses (start an inner portion of a formula).
- # Value at cursor location.

A formula is a value that consists of one or more of the following:

- A number: 1, -11, 25.5, 3.4E4.
- A reference to the coordinates of another location: A5, J23.
- A range of such references: B2...B2, F5...F0.
- An arithmetic operator: +, -, =, \*, /, ^.
- A function: @SUM(M2...M2), @PI (all functions begin with @).

All typed references to other cells or formulas must be entered by typing + before the first letter of the reference (+ B2); otherwise the program accepts the reference or formula as a label (B2).

When the first character of an entry is recognized as a value, the word *Value* appears on the prompt line, and the typed character appears on the edit line. Each subsequent character appears on the edit line as it is typed until the value is written on the worksheet (by pressing RETURN or an arrow key) or canceled (by pressing CONTROL-C).

If you set the Mode option of the Attribute command to restrict cell contents to values only (/AMV), you do not need to type + to begin a formula whose first character is a cell coordinate (e.g., A1/B1-C1). If you point to the cell coordinate with the cursor, however, you must type the + first. See the section on "The Attribute Command" in this chapter for more information on /AMV.

If you make a typing error while typing a value, press ESCAPE or CONTROL-E to erase the mistake, and type the correct character(s). The Edit command (/E) can be used to correct a value after it is written on the worksheet.

When a formula is written on the worksheet, its calculated value is displayed. The formula itself appears on the entry line of the status area. You can see the formula in its cell by typing /AEY.

## **References to Other Cells**

The value of another cell can be used in a formula by including the coordinates of the other cell. The value changes whenever the entry in the other cell changes.

When the first element of a formula is a reference to another cell, the formula must begin with a character that starts a value such as +. If it begins with a letter, the VisiCalc program assumes the entry is a label with a value of zero (see "Labels" in this chapter).

If the last character typed is an arithmetic operator (+, -, =, \*, /, <, >), typing an arrow key does not complete the formula nor write it on the worksheet. The formula can be completed by pointing the cursor to another cell and pressing RETURN. When you point the cursor to another cell, the edit line displays the coordinates of the cell. Pressing RETURN returns the cursor to the cell with the original formula, accepts the value of the cell to which the cursor was pointing, writes the coordinates of the cell into the formula, and writes the formula onto the worksheet.

For example, if 1 + is on the edit line and the cursor is moved to A5, the formula on the edit line is 1 + A5. Typing an arrow key at this point does not write the formula on the worksheet; it moves the cursor and changes A5 to the new coordinate of the cursor. RETURN must be typed to complete the formula.

If the formula is to be 1 + A5 - B5, typing - after moving the cursor to A5 returns the cursor to the formula cell. Now the cursor can be moved to B5, which makes the formula on the edit line 1 + A5 - B5; RETURN completes the formula.

The VisiCalc program does not allow an illegal formula to be entered (such as one that ends with an arithmetic operator). It beeps an error and waits for the formula to be completed. To continue after an illegal entry of this type, the formula must be edited or canceled. A valid formula that creates an illegal calculation (such as dividing by zero) displays the special value ERROR at the formula cell and at all cells that reference the formula cell.

The complexity of a formula (the number of references to other locations, arithmetic operators, parentheses, and memory available) determines the maximum length of a formula. If a formula becomes too complex, the VisiCalc program beeps an error and stops displaying additional characters on the edit line. Only the formula up to that point can be written on the worksheet.

## **Precision of VisiCalc Values**

The VisiCalc program stores all values in base 10 with either 11 or 12 significant digits. The largest number the VisiCalc program can accurately calculate is .9999999999999E62. The smallest is 9.9999999999E-66. The E62 means "times 10 to the 62nd power," and E-66 means "times 10 to the negative 66th power." When a number is displayed in the General format (see "Format Command" in this chapter), the VisiCalc program shifts between conventional and scientific notation as required to display the value with the greatest precision.

In scientific notation with a column width of 9, the number 1234568923 is displayed as 1.235E10. Non-significant zeros are dropped, except when fixed-display formats and attributes, such as Dollars-and-Cents format, are in effect.

If the columns are too narrow to display numbers even in scientific notation, the VisiCalc program fills its cells with > signs. The display format (Integer, Dollars-and-Cents, etc.) and column width do not change the number stored in memory, only the way it is displayed.

## **Order of Calculation in Formulas**

The VisiCalc program performs formula calculations in the order it encounters each operator from left to right. No operator takes precedence over another. However, parts of a formula in parentheses are calculated first. If there are parentheses within parentheses, the VisiCalc program calculates the innermost first. For example,  $5 + 6/2*4$  evaluates to 22, but  $5 + ((6/2)*4)$  evaluates to 17.

## **Calculating Values on the Edit Line**

You can do separate calculations on the edit line as you build a worksheet. The edit line becomes a calculator; any valid formula can be typed and evaluated using the !, then cleared with CONTROL-C.

Typing an exclamation point (!) at the end of a formula causes the calculated value to replace the formula on the edit line. Typing a crosshatch (#) immediately after a reference to another location causes the value of that location to replace the reference to it on the edit line. Typing percent (%) causes the value of the cell to be divided by 100 and displayed on the edit line. If you press RETURN, the value divided by 100 is entered in the cell.

The following example shows the # and ! features.

Example:

<b>Type:</b>	<b>Result:</b>
1. /CAY	The worksheet clears and the cursor moves to A1.
2. 1 → 2 → 3 ← ←	Three values in row 1.
3. ↓ + ↑ + ↑ · → + ↑ → →	Entry Line: A2 Edit Line: +A1+B1+C1
4. #	Edit Line: +A1+B1+3  The reference to C1 is replaced by the value of C1. If RETURN is pressed now, the formula on the edit line is written at A2.
5. !	Edit Line: 6  The entire formula on the edit line is replaced by the calculated value of the formula. If RETURN is typed now, the value 6 is written at A2.
6. CONTROL-C	Entry Line: clear Edit Line: clear A2: clear

The entry is canceled.

The # can be used to copy the number (not the formula) from one cell to another. With the cursor at the destination cell, type +, move the cursor to the cell that is to be copied (or type its coordinate), and type # RETURN. The value of the cell is calculated and written at the destination. The value of the destination cell does not change when the value of the origin cell changes because the value of the origin was copied, not the coordinate.

## **Recalculation Order**

Recalculation order determines in what order the cells on the worksheet are calculated. This has implications for forward references and the accuracy of the calculated values on a worksheet.

A formula can have as many references to other locations as the complexity of the formula allows. Unless manual recalculation is specified (with /GRM), the worksheet is recalculated each time an entry is written on the worksheet. Recalculation always starts at location A1.

When first loaded, the VisiCalc program is set to calculate column-by-column, from left to right. It calculates the value at A1, then A2, then A3 to the end of column A. Then it calculates B1, B2, B3 to the end of column B; then C1, C2, C3 and so on. The C in the upper-right corner of the screen indicates that the order of recalculation is by column.

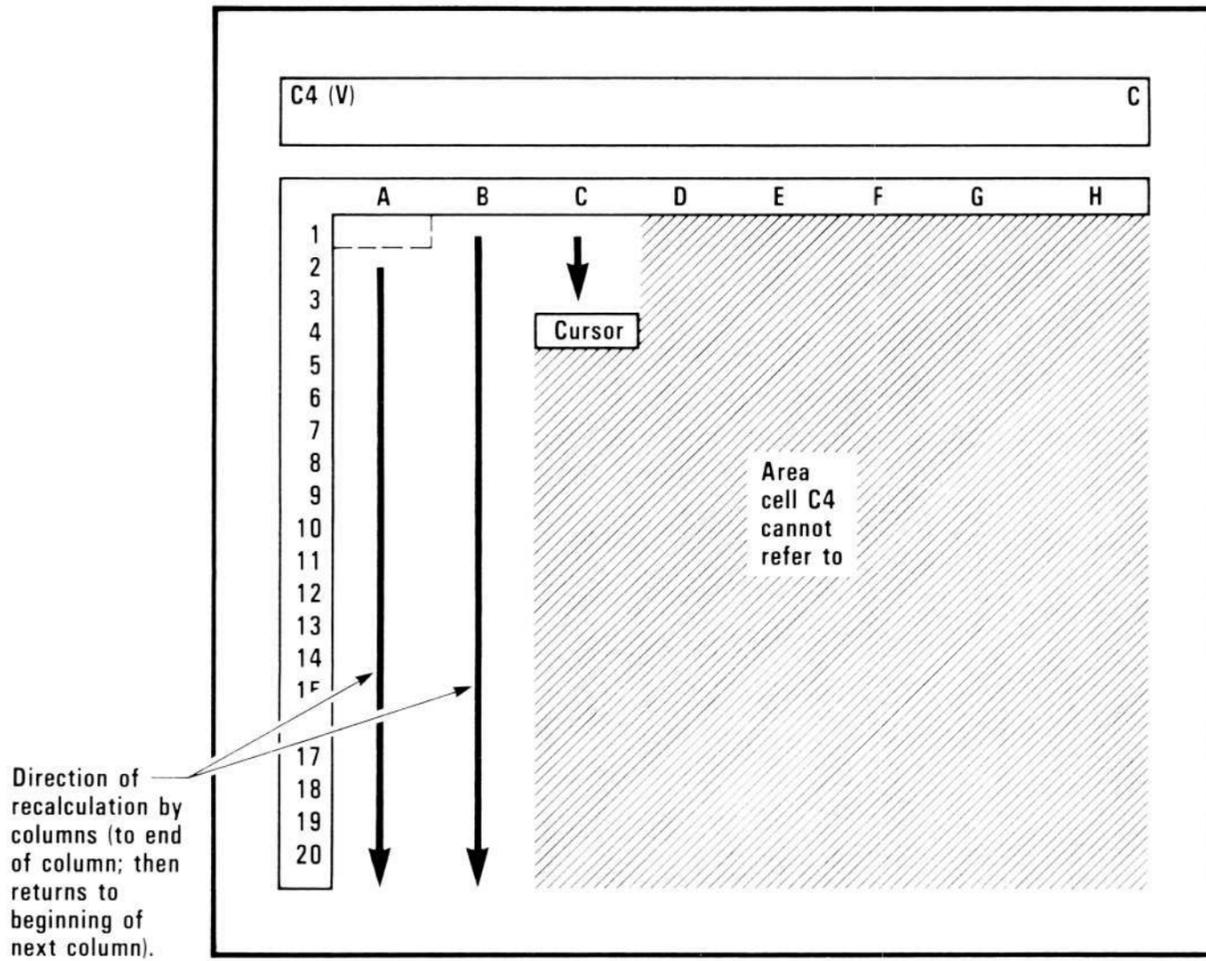
The Global command (/GO) changes the order of calculation from column-by-column to row-by-row and changes the C in the upper-right corner of the status area to an R. Row 1 is calculated from left to right, starting at A1; row 2 is calculated from left to right, starting at A2, and so on.

## **Forward and Circular References**

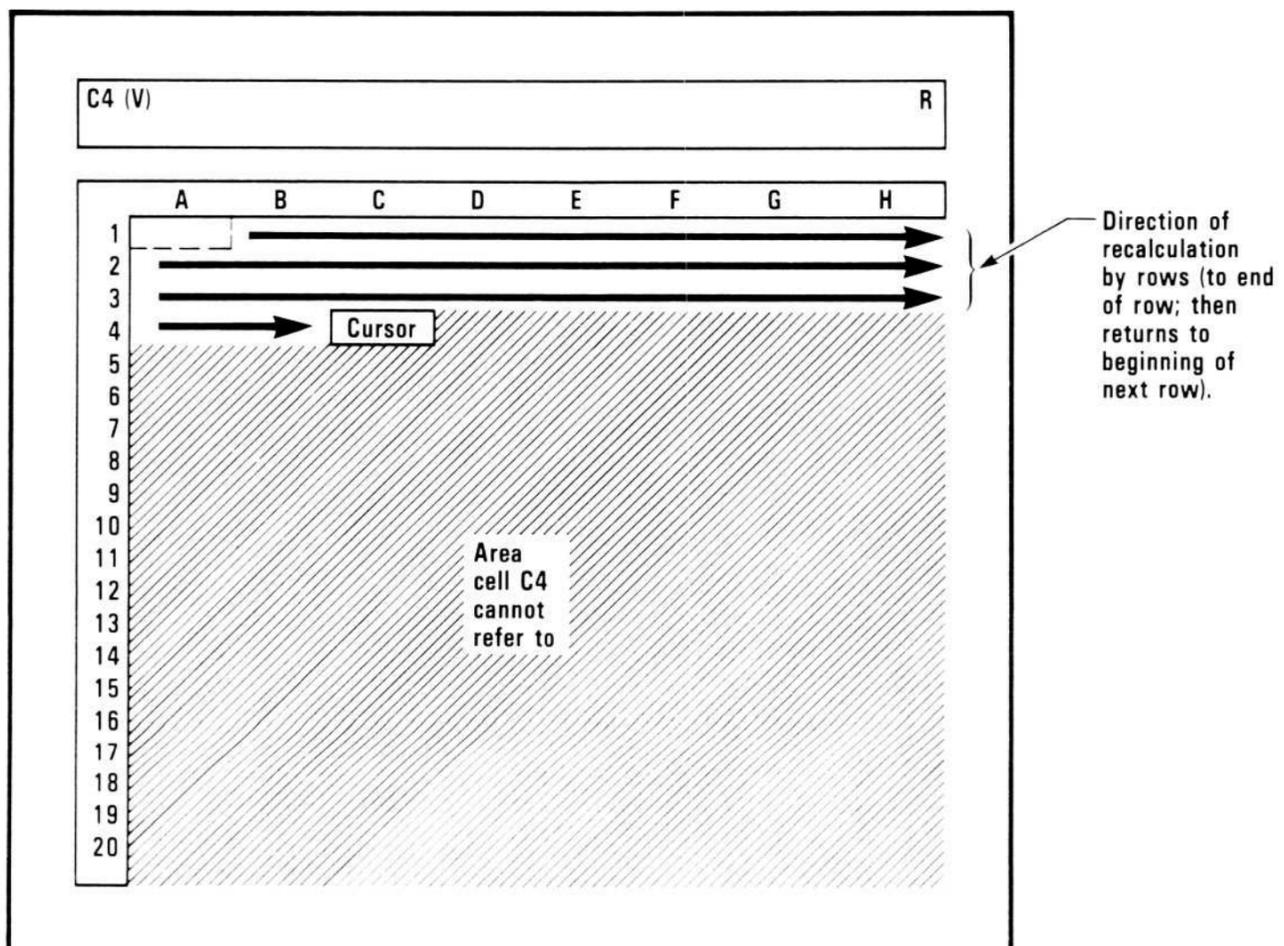
The worksheet formulas should be arranged so that all formulas with references to other cells are located *after* the referenced cells are calculated. When the VisiCalc program recalculates by columns, the cursor cell should refer only to cells in columns to the left or in the same column above the cursor cell. When the VisiCalc program recalculates by rows, the cursor cell should refer to cells in rows above or in the same row to the left of the cursor cell.

Figure 3-33 (a and b) illustrates forward references.

a) Forward references when recalculating by column.



b) Forward references when recalculating by row.



010-035

Figure 3-33 a, b. Forward References by Column and Row

If the worksheet formulas are not arranged in this way, a formula containing a reference to another cell is recalculated before the new value of the referenced cell is calculated. After the worksheet is recalculated, it displays the *old* value of the referenced cell. The referenced cell itself, however, displays its *new* value. This condition is a forward reference.

Such *forward references* are often difficult to diagnose. If you suspect the worksheet contains a forward reference, type !. This forces another recalculation. Sometimes it is necessary to type ! several times before the worksheet is correct.

Watch the cell whose value is incorrect. If its value changes, look for forward references; the worksheet should be redesigned to eliminate them. (It is possible to not change the forward reference and type ! to force an extra recalculation; but this can be inconvenient, especially if there are forward references to other forward references, a situation that requires several forced recalculations to produce a correct result.)

Forward references are frequent when column totals are placed at the *top* of the page.

**NOTE:** If a worksheet that includes forward references is saved and later loaded from diskette, cells containing forward references display **ERROR**. A recalculation must be forced (by typing !) to eliminate the **ERROR** values and display correct results. The forward references should be eliminated by reorganizing the worksheet.

A *circular reference* is a formula that refers to its own cell, such as  $1 + A1$  in cell **A1**. Each time the worksheet is recalculated, the value of this formula changes even if no other change is made to the worksheet. A circular reference prevents accurate results and is particularly difficult to diagnose.

**VERSION COMMAND—/V**

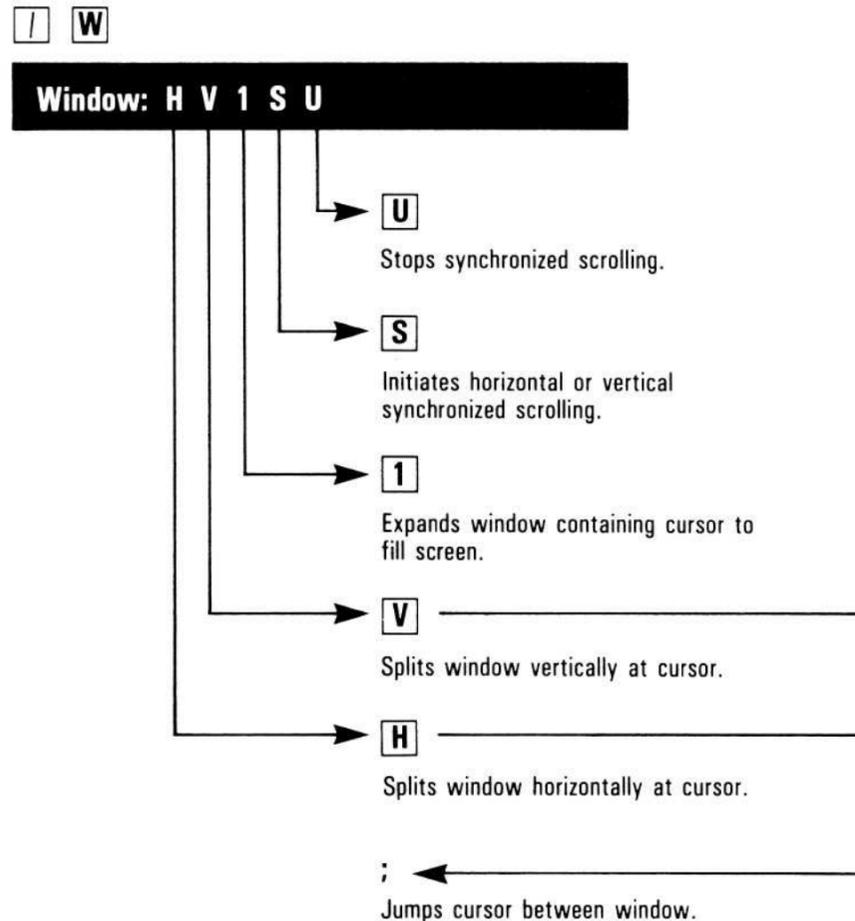
| V

© 1982 Software Arts, Inc. X.XX

Displays copyright notice and version number on prompt line, clears automatically with next keystroke.

The Version command displays in the status area the version number of the VisiCalc program. It does not affect the worksheet. To enter the Version command, type /V. The version number identifies your VisiCalc program and must accompany any inquiries to the dealer or manufacturer.

## WINDOW COMMAND—/W



The Window command (/W) splits the screen vertically or horizontally so that the worksheet can be viewed through two windows simultaneously. Two windows can display rows or columns that are too far apart to be viewed through a single window. You can not print a split worksheet as it appears on your screen.

When the screen is split, the windows are scrolled independently or together. Worksheet cells can be displayed in different formats in each window. To move to the other window, type ;.

Any VisiCalc command can be entered in either window. With three exceptions, the effect of a command on the worksheet is shown in both windows. The three exceptions are the Column, Attribute and Format options of the Global command, which are set in one window at a time. Single column widths are unaffected by the Window command.

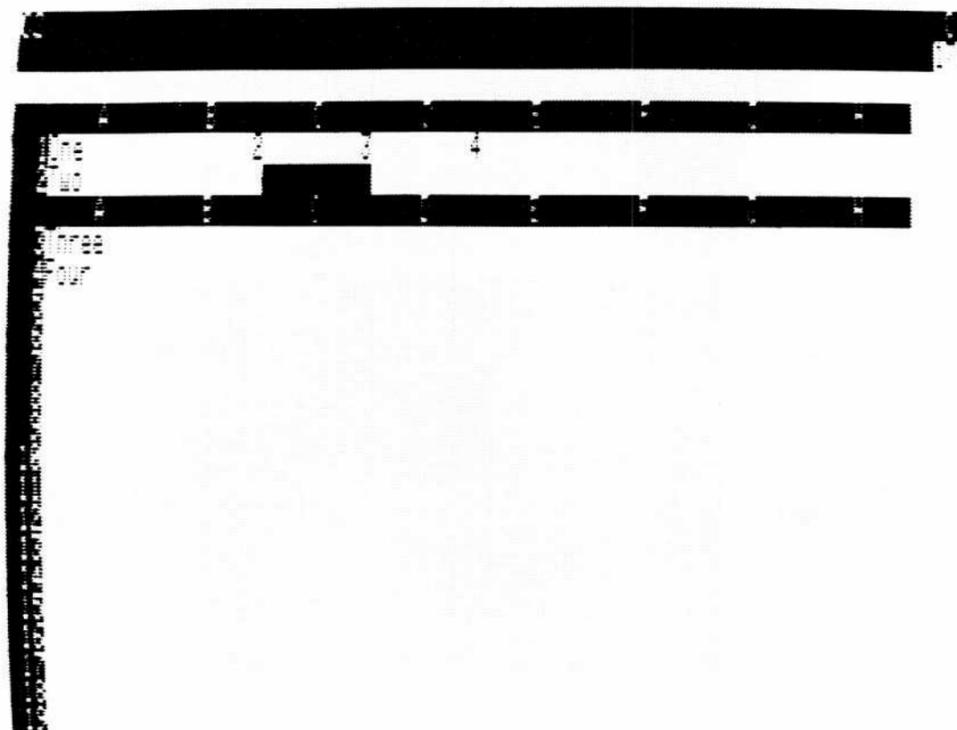
### Horizontal Option—/WH

The Horizontal option of the Window command (/WH) splits the window just above the row that contains the cursor. A horizontal border separates the two screens. When the command is executed, the cursor moves up one row into the upper window. For example, typing /WH when the cursor is at A4 places the horizontal border between A4 and A3 and moves the cursor to A3.

/WH cannot work if the cursor is in row 1 —there's no "between" and nowhere to put the cursor. The following example shows the horizontal window option:

Example:

<b>Type:</b>	<b>Result:</b>
1. /CAY	The screen clears and the cursor moves to A1.
2. → 2 → 3 → 4 RETURN	A row of numbers in row 1.
3. >A1 RETURN	
4. One ↓ Two ↓ Three ↓ Four ↓	A column of words in column A.
5. >C3 RETURN	Cursor moves to C3.
6. /W	Prompt Line: Window: H V 1 S U
7. H	The screen should look like Figure 3-34. (note the position of the cursor and the horizontal bar).



010-037/P

Figure 3-34. Horizontal Windows

Each window now shows the worksheet independently. The cursor can move around the worksheet in the top window. Move the cursor to A1 to highlight One in the top window. Type ; to move the cursor to the bottom window. Typing ; moves the cursor to its last position in the other window. In this case, that cell was C3. Type ; again. The cursor returns to A1, the last cell it rested on before it moved to the lower window.

Move to the bottom window with ; and remove the horizontal window by typing /W1. There is now one window, the window where the cursor was when /W1 was typed. This window has the format options of the window that contained the cursor when the Window command was entered.

The screen can be split into two windows only when one window is displayed (you can't change directly from horizontal windows to vertical windows or vice versa). The size of each window is determined by the position of the cursor at the time the Window command is entered.

### **Vertical Option—/WV**

The Vertical option of the Window command (/WV) splits the window vertically to the left of the column that contains the cursor. A vertical border separates the windows. For example, typing /WV when the cursor is at B3 places the vertical border between A3 and B3 and moves the cursor left to A3.

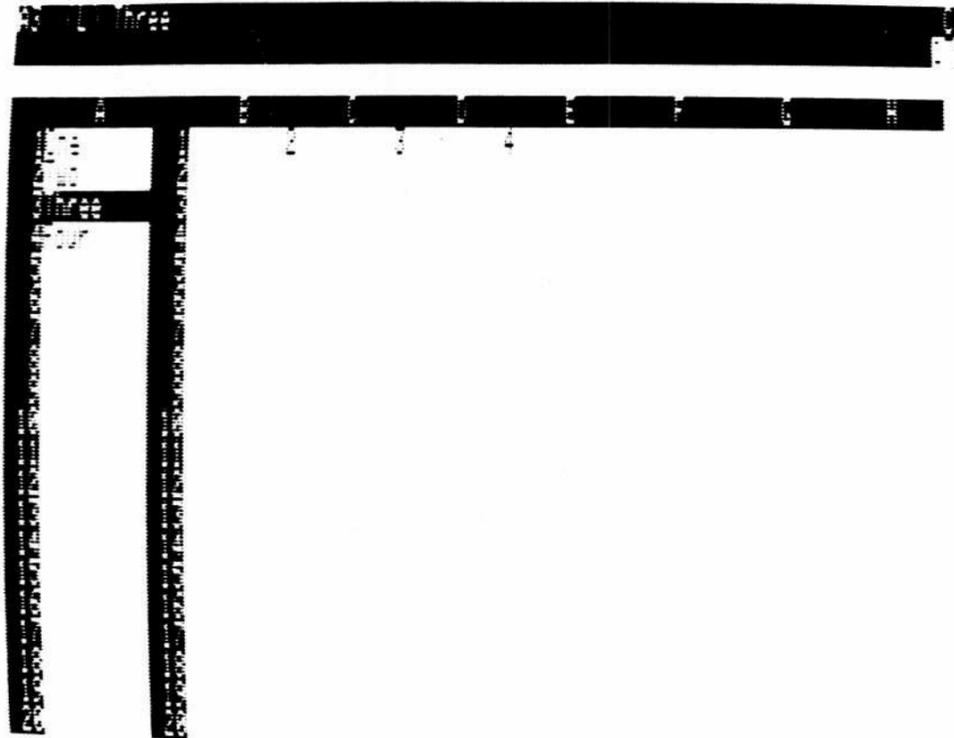
To make room for the additional row border, the columns in the right window might be slightly narrower than those in the left window. As a rule, the VisiCalc program narrows the right window by one space if the narrowing preserves a column on the screen.

The behavior of vertical windows—cursor movement, effect of commands, formats, etc.—is the same as horizontal windows.

If the previous example is not on the screen, type steps 1 through 4 from the horizontal window example before continuing.

Example:

Type:	Result:
1. > B3 RETURN	The cursor moves to B3.
2. /W	Prompt Line: Window: H V 1 S U
3. V	The screen should look like the Figure 3-35 (note the position of the border and cursor).



010-038/P

Figure 3-35. Vertical Windows

### **One Window Option—/W1**

The One Window option of the Window command (/W1) displays a single window. If the command is entered while the screen is split, the global format and column width settings of the window that contains the cursor apply to the single window.

### **Synchronized Scrolling Option—/WS**

The Synchronized Scrolling option of the Window command (/WS) synchronizes horizontal scrolling in horizontal windows or vertical scrolling in vertical windows. If the window that contains the cursor is scrolled, the other window also scrolls.

### **Unsynchronized Scrolling Option—/WU**

The Unsynchronized Scrolling option of the Window command (/WU) cancels synchronized scrolling.

**NOTE:** The last three options of the Window command (/W1, /WS, and /WU) can only be used when the screen is split.



---

**VISICALC® Advanced Version****OPERATING THE APPLE III****WHAT YOU NEED**

The VisiCalc® Advanced Version program runs on the Apple III computer with at least 128K memory and the built-in disk drive. At least one external drive or hard disk is required. You can use either a floppy disk or a hard disk for storing your worksheets. You'll need a printer, unless you don't want to print your worksheets.

You also need the VisiCalc Advanced Version Loader and Program diskettes. If you want to use a floppy disk drive for storing your worksheets, you'll need at least one initialized diskette. The diskette **must** be initialized **before** you load the VisiCalc (AV) program. See the *Apple III Owner's Guide* for information on how to initialize a diskette. If you want to use a hard disk unit for worksheet storage, create at least one subdirectory for that purpose. See the section "Make a Subdirectory" in Chapter 5 of the *Apple III Owner's Guide* for directions.

**THE KEYBOARD**

Figure A-1 shows the keyboard. Most of the keys are self-explanatory, but a few need further description. Note the keys that are shaded; you will use them a great deal.

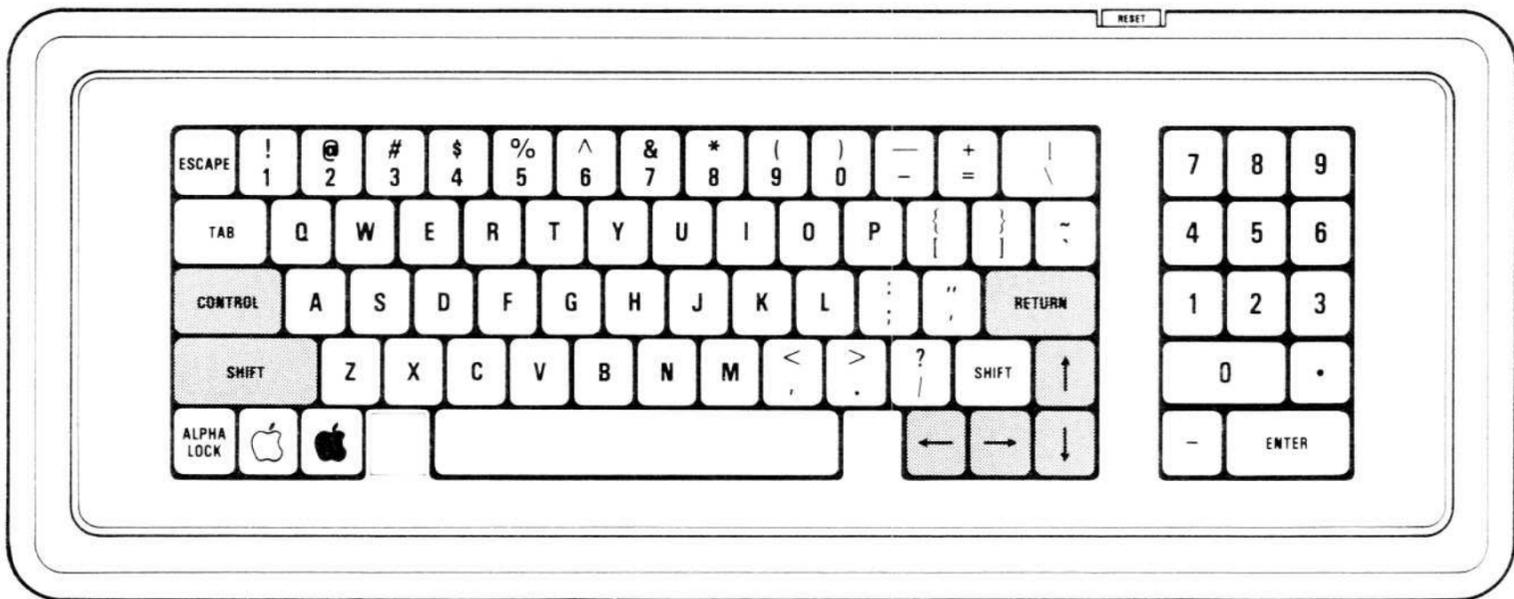
The keys marked with arrows move the cursor on the screen.

The RETURN key ends commands and worksheet entries.

The ESCAPE key backspaces to correct typing errors.

Holding down the CONTROL key and typing C cancels an entire entry or command; it also stops the printer or diskette drive when printing or saving a worksheet.

The Shift key works just like it does on a typewriter.



009-001

Figure A-1. Apple III Keyboard

## HANDLING DISKETTES

A diskette is a small, magnetically coated plastic disk, sealed in a protective square plastic cover. Through the oval cutout in the square cover, you can see the magnetic surface of the actual diskette.

*Never* touch the exposed magnetic surface with your fingers or any implement; you could lose data. Protect the diskette from dust by storing it in the paper sleeve it comes in. Keep it at least six inches from magnetic fields such as those generated by a TV. Extremes of temperature (such as in a car trunk on a warm day) can damage a diskette, destroying valuable data or your VisiCalc program. Don't bend, staple, or write on the square plastic cover with a hard pen or pencil (use only a soft felt-tip pen).

## LOADING THE VISICALC PROGRAM

First, make sure you have at least one freshly initialized diskette available. Follow the directions under "Formatting Blank Diskettes" in the *Apple III Owner's Guide*. If you are using a hard disk unit for storing your worksheets, create a subdirectory on the hard disk. See the section "Make a Subdirectory" in Chapter 5 of the *Apple III Owner's Guide* for directions.

As you face the front of the computer, the power switch is at the rear of the left side. With the power switch off, open the built-in diskette drive door. If you are using an external floppy disk drive for storing your worksheets, open that disk drive door, too. If a diskette is in either drive, remove it. With the drive door(s) open and the power off, follow this procedure:

1. Remove the VisiCalc Loader diskette from its paper sleeve and hold it by the label edge with the label side up. Carefully insert the diskette into the built-in drive with the label edge entering last (see Figure A-2). Gently push the diskette all the way in and close the drive door.

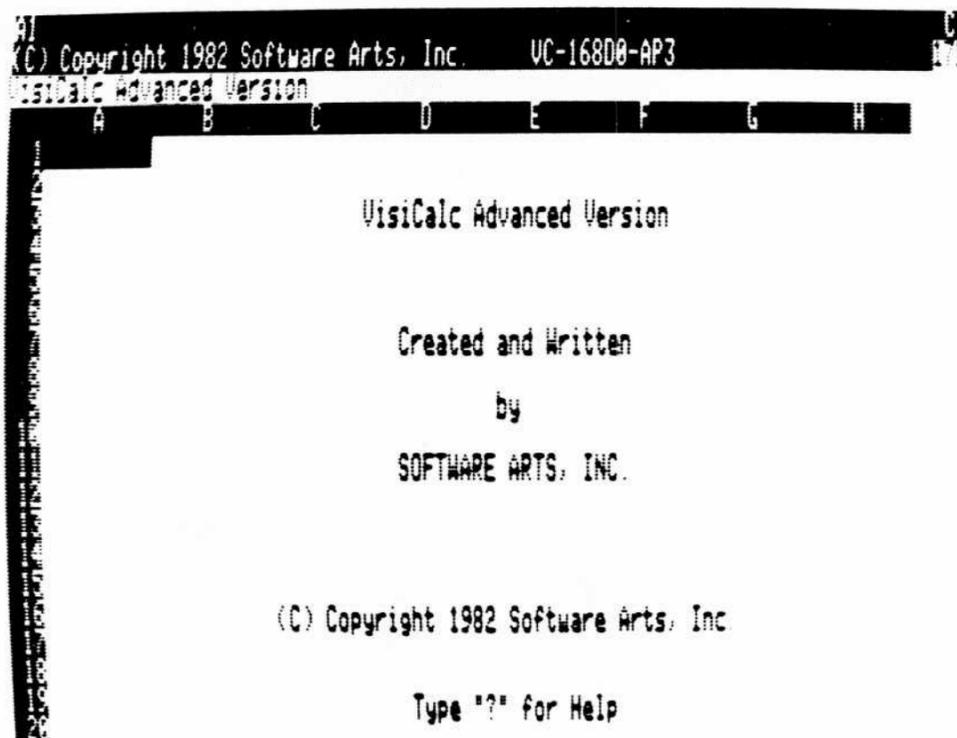


009-001/P

**Figure A-2. Inserting a Diskette**

2. Turn on the power switch. The drive motor whirs, and a copyright notice appears on your screen. Then the VisiCalc screen appears. The prompt line reads :Program file missing. Switch disks and press RETURN.
3. Remove the VisiCalc Loader diskette from the built-in drive and put it back in its paper sleeve. Repeat step 1 with the VisiCalc Program diskette.

4. Press the **RETURN** key. After a few seconds, the VisiCalc copyright information appears on the screen. The screen should look like Figure A-3.



010-036/P

**Figure A-3. The VisiCalc Screen**

If the screen doesn't look like Figure A-3, remove the VisiCalc Program diskette from the drive. Turn off the computer and go through the load procedure again. If, after several tries, the VisiCalc program does not load, see your dealer for help.

5. After you have loaded the program successfully, keep the VisiCalc Program diskette in the built-in drive until you are finished using the program.
6. If you are using an external floppy disk drive, insert the initialized diskette into that drive and close the door. You are ready to use the VisiCalc program.

---

**VISICALC® Advanced Version**

## CONFIGURATIONS FOR THE VISICALC PROGRAM

Every diskette you use to start up (boot) your Apple III must contain software programs called device drivers. The drivers enable your Apple to send data to and from disk drives, printers, the console, and other peripheral devices.

The drivers are in a file called SOS.DRIVER in each boot diskette. A diskette will not boot the Apple III if it does not contain a properly configured SOS.DRIVER file. The SOS.DRIVER file must contain a device driver for each peripheral you want to use.

The SOS.DRIVER file on the VisiCalc boot diskette (the Loader diskette) is supplied with device drivers to support:

- The built-in screen and keyboard (.CONSOLE);
- The Apple III's built-in speaker (.AUDIO);
- An Apple Silentype™ printer (.PRINTER);
- A Qume™ Sprint 5/45 printer (.QUME);
- A ProFile hard disk (.PROFILE).

The driver parameters are set up to allow for the built-in disk drive and an external floppy disk drive.

Using the System Configuration Program (SCP) on your Apple III Utilities diskette, you can reconfigure your VisiCalc diskette in order to:

- Change the number of disk drives;
- Add other printer drivers or modify a driver for another printer;
- Add other peripheral devices.

To reconfigure your diskette, boot the Apple III System Utilities diskette. Use SCP to edit the system configuration on your VisiCalc Loader diskette. See Chapter 2 of your *Apple III Standard Device Drivers Manual* for information on how to do this.

Be very careful when reconfiguring your VisiCalc Loader diskette! If you don't do it properly, you may render your VisiCalc diskette unusable—and because it's copy-protected, you can't make a backup copy first to protect yourself. Follow the directions on SCP in the *Apple III Standard Device Drivers Manual* closely and carefully.

For the VisiCalc program to operate properly, the system configuration on your VisiCalc diskette must contain at least the console and audio drivers. You may include any number of other drivers, but if they are active, they will take up some of the memory available to the VisiCalc worksheet. For this reason, we recommend that you have active only the drivers that you need to use the VisiCalc program. This will give you the maximum amount of space in the VisiCalc worksheet. For further information on active and inactive drivers and how to activate and deactivate them, see the section "Driver Status" in Chapter 2 of the *Apple III Standard Device Drivers Manual*.

The driver for the printer you wish to use as your standard printer should be named `.PRINTER`. The driver for the Silentype is named `.PRINTER`, but you can specify another printer by first deleting the `.PRINTER` driver and loading the new printer driver and renaming it `.PRINTER`. This is the driver that will be used when you type the VisiCalc command `/PP`. To learn how to change a driver name, refer to the section "Editing Driver Parameters" in Chapter 2 of the *Apple III Standard Device Drivers Manual*.

You may also include other printer drivers in your VisiCalc system configuration with different names. If they are active, you can use them for printing by typing `/PF.NAME`, where `NAME` is the name of the printer driver you wish to use.

The following printers were tested and performed satisfactorily with the program:

- Apple® Silentype™
- Centronics® 737 and 739
- IDS Paper Tiger™ 440, 445, 460, and 560
- Epson MX-80 and MX-100
- NEC Spinwriter™ 5520 and 5530
- Qume® Sprint 5/45 and 9/55

These printers were tested with the port (Port A or Port C, as appropriate) or interface card shown in Table B-1. An X in the table means the combination was tested satisfactorily, an F means it failed the test, and a blank means it was not tested.

**Table B-1. Printer and Interface Card Tests**

<b>Printer</b>	<b>Port A</b>	<b>Port C</b>	<b>Universal Parallel Interface Card</b>
Apple Silentype	X		
Centronics 737			X
Centronics 739			X
IDS Paper Tiger 440		X	X
IDS Paper Tiger 445		X	X
IDS Paper Tiger 460		X	X
IDS Paper Tiger 560		X	X
Epson MX-80			X
Epson MX-100			X
NEC Spinwriter 5520		X	
NEC Spinwriter 5530			X
Qume Sprint 5/45		X	
Qume Sprint 9/55		X	

X = Tested with satisfactory results.

F = Tested and does not operate with the program

= Not tested

## COPYING PROGRAM FILES TO YOUR PROFILE™ HARD DISK

If you have a ProFile™ hard disk, you can copy the files on your Program Diskette to your hard disk, which speeds up overlay and help-text loading. This also saves time when you boot the program, because you do not have to switch diskettes. To make VisiCalc work on your hard disk, follow these steps:

1. Check to make sure that your SOS.DRIVER file includes the .PROFILE driver (see preceding section for details).
2. Boot your Apple III Utilities Diskette, and select File Handling from the main menu.
3. When the File Handling menu appears, select M-Make a New Sub-directory.
4. An entry prompt appears inside brackets at the bottom of your screen. Type **.profile/avc2 RETURN**.
5. When the prompt with enough space for [25] files appears, type **05 RETURN**. The red light on your hard disk will blink as the sub-directory is being created.
6. Press **ESCAPE** to get back to the File Handling menu.
7. Select C-Copy Files from the File Handling menu.
8. If you have two floppy disk drives, put your Program Diskette in drive 2, and type **.d2/=RETURN**. If you have only one floppy disk drive, take out the utilities disk, put in your program disk, and type **.d1/=RETURN**.
9. When the copy files to prompt appears, type **.profile/avc2/=RETURN**.
10. The files VISICALC.OVL, HELP.TEXT.FILE, INVOICE2.VC and INVOICE3.VC should now be on your hard disk.

**NOTE:** Remember that your tutorial worksheet files (Invoice2.VC and Invoice3.VC), which allow you to start lessons 5 and 6 without first doing lessons 4 and 5, are in the AVC2 subdirectory.

11. Load the VisiCalc program. With the cursor in cell A1 of the empty worksheet, type **“.profile/avc2 RETURN**.

12. Pick up the Loader diskette. On the right edge of the diskette, near the top, is a thin piece of plastic with an adhesive backing, called a *write-protect tab*. (It looks like a piece of electrical tape.) Carefully peel the tab off and put it on your desk or someplace handy. Don't throw it away or stick it to anything; you will replace it on the diskette in just a minute. Try not to bend the diskette when peeling off the tape; you could damage it if you are not careful.
13. Type **/S#S .dl/vcswappath.dif**. Do not type RETURN yet.
14. Take the Program diskette out of the built-in drive (drive 1).
15. Put the Loader diskette (with the write-protect tab removed) into the built-in drive (drive 1) and close the drive door.
16. Press **RETURN**. The program prompts for the lower-right coordinate of the data area to be saved; type **A1 RETURN**. The program will respond with Program file missing. Switch disks and press RETURN. Remove the Loader disk and put the Program disk into the built-in drive. Press **RETURN**. The next prompt will ask you to Restore user disk to drive; remove the Program disk and replace the Loader disk. Press **RETURN**. The last prompt will read Data save: R C or RETURN. Press **RETURN**. The DIF file will be saved on your Loader diskette.
17. Carefully replace the write-protect tab on the Loader diskette.
18. Put the Program diskette back in the built-in drive (drive 1).

Now all you have to do to load VisiCalc is turn on your ProFile™ and put the Loader diskette in drive 1. The program automatically looks for the overlay files on the ProFile.



**VISICALC® Advanced Version****EXCHANGING FILES WITH DIF™ (DATA INTERCHANGE FORMAT)**

The DIF™ format is a standard file format that allows programs to share data.

- Within the VisiCalc program, you may use it to manipulate sections of the worksheet and to transfer sections of one worksheet to another. This allows data consolidation for purposes such as corporate or annual calculations. You can use the DIF format this way, regardless of previous computer experience.
- The VisiCalc program also saves data in text files that can be read by other DIF-supporting programs. Thus, a file created by the VisiCalc program can be saved and read by other programs that support DIF. Files created by other programs can be loaded by the VisiCalc program. These programs can be written (in BASIC, for example) so that data generated by the VisiCalc program can be integrated into a broader set of personal computing tools. Use of the DIF format in this way is recommended for experienced programmers only.

This appendix describes the VisiCalc commands that save and load DIF files and how the files can be used within the VisiCalc program. It then describes the format itself and lists three sample programs in BASIC that read and write DIF files.

Additional information about DIF™ is available through the DIF Clearinghouse. The DIF Clearinghouse is set up to:

- Coordinate and distribute information about DIF format.
- Maintain and distribute DIF Technical Specification.

Information about the DIF Clearinghouse can be obtained by writing to:

DIF Clearinghouse P.O. Box 527 Cambridge, MA 02139

DIF™ is a trademark of Software Arts, Inc.

## THE SAVE DIF™ OPTION — /S#S

The Save DIF option of the Storage command (/S#S) saves the worksheet in the DIF format. You may want to add the extension “.DIF” to any DIF files you save so they can be recognized later as DIF files.

The VisiCalc DIF option saves and loads a worksheet in one of two ways—by row or by column. The VisiCalc program prompts you to specify the orientation by typing **R** or **RETURN** or **C**. **R** or **RETURN** save worksheets by rows; **C** saves by columns. Specify **R** or press **RETURN** when saving a worksheet to be loaded later by the VisiCalc program.

When a worksheet is saved in the DIF format, all formulas are converted to their resultant values. Repeating labels are converted to the initial label without repetitions.

## LOAD DIF™ OPTION — /S#L

The Load DIF option of the Storage command (/S#L) loads a worksheet stored on a diskette in the DIF format. The file can be one saved with the Save DIF option of the Storage command, or created by another program.

Unlike a worksheet saved in the normal fashion—which is loaded into the same position it occupied when it was saved—the upper-left corner of a worksheet loaded in the DIF format is the position of the cursor when the Storage command is typed.

As with the other options of the Storage command, you can display the names of the files on the diskette by typing → in response to the File to load: prompt.

The VisiCalc DIF option loads a worksheet in one of two ways—by row or by column. The VisiCalc program prompts you to specify the orientation by typing **R** or **RETURN** or **C**.

## USING THE DIF™ FORMAT WITHIN THE VISICALC PROGRAM

The DIF format can be used to save rectangular portions of the worksheet and load them back in another place on the same or a different worksheet. The cursor position indicates the upper-left coordinate of the portion to be saved. You are prompted to enter the lower-right coordinate. When you reload the file, the upper-left corner of the saved portion is placed at the current cursor position.

Saving files in the DIF format converts all formulas to values. If you have a section of a worksheet containing formulas and wish to convert the formulas to values, save that section in the DIF format with a new file name and reload it again in the same place on the worksheet. All repeating labels will be converted to the initial labels without repetitions.

The DIF format allows transposing of rows and columns. A worksheet or section of a worksheet might be saved in the DIF format with R or RETURN to specify row orientation. When the worksheet is loaded, column orientation can be specified. The rows will then be displayed as columns and the columns as rows. Likewise, data may be saved by columns and loaded in rows.

For example, this file,

A1	B1	C1
A2	B2	C2
A3	B3	C3

is saved by rows in the DIF format with A1 as the upper-left coordinate and C3 given as the lower-right coordinate. When the file is loaded with the cursor at A1, column orientation is specified. The result is:

A1	A2	A3
B1	B2	B3
C1	C2	C3

## THE DIF™ FORMAT

The following information is provided for experienced programmers.

The DIF format stores the worksheet in a form accessible to programs other than the VisiCalc program. To accommodate a wide range of languages in which such a program might be written, several simplifying techniques have been used:

- Information about the size of the file is provided at the beginning.
- Records are kept as short as possible.
- The data type (string or number) of each value is explicitly defined.
- Strings are stored one per line.
- Strings that contain special characters are enclosed in quotation marks.
- The file ends with an explicit End-Of-Data record.

The following table shows a sample worksheet used to describe the format and the programs that work with the DIF format.

<b>Year</b>	<b>1980</b>	<b>1981</b>	<b>1982</b>
Sales	100	110	121
Cost	80	88	97
Profit	20	22	24

### Sample Worksheet

The format stores the worksheet in a series of slices; the worksheet can be sliced either horizontally (by rows) or vertically (by columns). Each of these slices is called a tuple; each slice along the other axis is called a vector. In the following table, for example, if the worksheet is saved by rows, the first vector is Year 1980 1981 1982 and the first tuple contains Year, - (one hyphen), Sales, Cost, and Profit; the entire worksheet is stored in four tuples of five values each.

If the worksheet in the following table is stored by columns, the first vector is Year – Sales Cost Profit and the first tuple contains Year 1980 1981 1982; the entire worksheet is stored in five tuples of four values each.

A DIF file consists of a series of *header* records that describe the file, followed by one set of *data* records for each tuple, and ends with a pair of *End-Of-Data* records.

**HEADER**

The header consists of four sets of three records that give information about the entire file:

TABLE
0,1
"
VECTORS
0,V
"
TUPLES
0,T
"
DATA
0,0
"

V is the number of vectors in the file.

T is the number of tuples in the file.

**DATA RECORDS**

The data records consist of a pair of header records that identify the beginning of a tuple, and a pair of records for each value in the tuple:

-1,0
BOT
T1,N1
string1
T2,N2
string2
T3,N3
string3
.
.
.
Tn,Nn
stringn

Beginning Of Tuple records.

First value of tuple.

Second value of tuple.

Third value of tuple.

Last (nth) value of tuple.

**END-OF-DATA RECORDS**

The End-Of-Data records mark the end of the file:

-1,0
EOD

## PROGRAMS THAT WORK WITH THE DIF™ FORMAT

The following three programs demonstrate how BASIC programs might be written and used with the DIF format. These are sample programs that must be typed in exactly or developed by programmers. These programs perform the following functions:

- Dump a DIF file just as it is stored, record by record.
- Print a worksheet from a DIF file.
- Create a DIF file by prompting for the worksheet entries.

### DUMPING A DIF™ FILE

This program prints the DIF file just as it is stored, record by record. It asks for the name of the file, and whether to print it. If not instructed to print the file, the program displays the file on the screen.

```

100 REM *****
110 REM *   INITIALIZATION   *
120 REM *****
130 NULS=CHR$(34)+CHR$(34)
140 FALSE=0
150 TRUE=1
160 PNTS=".SILENTYPE"
440 REM
450 REM
460 REM *****
470 REM *   MAIN ROUTINE   *
480 REM *****
490 REM
500 GOSUB 1000:REM ** PROMPT FOR ORDERS **
510 GOSUB 1200:REM ** PRINT HEADER **
520 GOSUB 1400:REM ** PRINT DATA RECORDS **
530 GOSUB 1600:REM ** END-OF-PROGRAM CLEANUP **
540 END
940 REM
950 REM
960 REM *****
970 REM *   PROMPT FOR ORDERS   *
980 REM *****
990 REM
1000 HOME
1010 INPUT"File name: ",FILENAMES
1020 IF RIGHTS(FILENAMES,4)<>".DIF" THEN FILENAMES=FILENAMES+".DIF"
1030 INPUT"Print the file (Y or N): ",REPLY$
1040 IF REPLY$="Y" OR REPLY$="y" THEN HARDCOPY=TRUE:OPEN#2 AS OUTPUT,PNTS
1050 OPEN#1 AS INPUT,FILENAMES
1060 IF NOT HARDCOPY THEN HOME
1070 IF HARDCOPY THEN PRINT#2;FILENAMES:PRINT#2;" ":PRINT#2;" ":ELSE PRINT FILENAMES:PRINT:PRINT
1080 RETURN
1140 REM
1150 REM
1160 REM *****
1170 REM *   PRINT HEADER   *
1180 REM *****
1190 REM
1200 INPUT#1;TITLES$
1210 INPUT#1;TYPE,NUMBER
1220 INPUT#1;STRNG$
1230 IF HARDCOPY THEN PRINT#2;TITLES$:PRINT#2;TYPE;",";NUMBER:ELSE PRINT TITLES$:PRINT TYPE;",";NUMBER
1240 IF STRNG$="" THEN IF HARDCOPY THEN PRINT#2;NULS:ELSE PRINT NULS:ELSE IF HARDCOPY THEN PRINT#2;STRNG$:ELSE PRINT STRNG$
1250 IF TITLES<>"DATA" THEN 1200
1260 IF NOT HARDCOPY THEN GOSUB 2000
1270 RETURN
1340 REM
1350 REM
1360 REM *****
1370 REM *   PRINT DATA RECORDS   *
1380 REM *****
1390 REM
1400 INPUT#1;TYPE,NUMBER
1410 INPUT#1;STRNG$
1420 IF VPOS>20 AND NOT HARDCOPY THEN GOSUB 2000
1430 IF HARDCOPY THEN PRINT#2;TYPE;",";NUMBER:PRINT#2;STRNG$:ELSE PRINT TYPE;",";NUMBER:PRINT STRNG$
1440 IF STRNG$<>"EOD" THEN 1400
1450 RETURN

```

## APPENDIX C

```

1530 REM
1540 REM
1550 REM *****
1560 REM *   END-OF-PROGRAM   *
1570 REM *   CLEANUP       *
1580 REM *****
1590 REM
1600 CLOSE#1:IF HARDCOPY THEN CLOSE#2
1610 RETURN
1920 REM
1930 REM
1940 REM *****
1950 REM *   PRINT           *
1960 REM *   "RETURN FOR MORE" *
1970 REM *   MESSAGE AT BOTTOM *
1980 REM *****
1990 REM
2000 VPOS=24:HPOS=1
2010 PRINT"Return for more";
2020 GET REPLY$:REM ** WAIT UNTIL ANY KEY IS PRESSED **
2030 HOME
2040 RETURN

```

## PRINTING A WORKSHEET FROM A DIF™ FILE

The following program prints the worksheet as it would appear on the screen. It asks for the name of the file in which the worksheet was saved in the DIF format, the width of columns to be printed, and whether the worksheet was saved by rows or by columns.

```

60 REM *****
70 REM *   INITIALIZATION   *
80 REM *****
90 REM
100 DIM WORKSHEETS(50,50)
110 FALSE=0
120 TRUE=-1
130 BYROWS=FALSE
140 PRNTS=".SILENTYPE"
440 REM
450 REM
460 REM *****
470 REM *   MAIN ROUTINE   *
480 REM *****
490 REM
500 GOSUB 1000:REM ** PROMPT FOR ORDERS **
510 GOSUB 1200:REM ** READ HEADER **
520 GOSUB 1400:REM ** READ DATA RECORDS **
530 IF NOT FILEBAD THEN GOSUB 1600:REM ** PRINT THE WORKSHEET **
540 GOSUB 1800:REM ** END-OF-PROGRAM CLEANUP **
550 END
940 REM
950 REM
960 REM *****
970 REM *   PROMPT FOR ORDERS *
980 REM *****
990 REM
1000 HOME
1010 INPUT"File name: ",FILENAME$
1020 IF RIGHT$(FILENAME$,4)<>".DIF" THEN FILENAME$=FILENAME$+".DIF"
1030 INPUT"Column width: ",COLUMNWIDTH
1040 INPUT"Saved by row or column (R or C): ",REPLY$
1050 IF REPLY$="R" OR REPLY$="r" THEN BYROWS=TRUE
1060 INPUT"Print the worksheet (Y or N): ",REPLY$
1070 IF REPLY$="Y" OR REPLY$="y" THEN HARDCOPY=TRUE:OPEN#2 AS OUTPUT,PRNTS
1080 OPEN#1 AS INPUT,FILENAME$
1090 RETURN
1140 REM
1150 REM
1160 REM *****
1170 REM *   READ HEADER     *
1180 REM *****
1190 REM
1200 INPUT#1;TITLES$
1210 INPUT#1;TYPE,NUMBER
1220 INPUT#1;STRNG$
1230 IF TITLES$="VECTORS" THEN VECTORS=NUMBER
1240 IF TITLES$="TUPLES" THEN TUPLES=NUMBER
1250 IF TITLES$="DATA" THEN RETURN
1260 GOTO 1200
1340 REM
1350 REM
1360 REM *****
1370 REM *   READ DATA RECORDS *
1380 REM *****
1390 REM

```

```

1400 FOR ROW=1 TO TUPLES
1410   INPUT#1;TYPE,NUMBER
1420   INPUT#1;STRNG$
1430   IF TYPE<>-1 OR STRNG$<>"BOT" THEN GOSUB 2000:RETURN
1440   FOR COL=1 TO VECTORS
1450     INPUT#1;TYPE,NUMBER
1460     INPUT#1;STRNG$
1470     IF TYPE<>0 AND TYPE<>1 THEN GOSUB 2000:RETURN
1480     IF BYROWS THEN IF TYPE=0 THEN WORKSHEET$(COL,ROW)=STR$(NUMBER):ELSE WORKSHEET$(COL,ROW)=STRNG$:ELSE IF
TYPE=0 THEN WORKSHEET$(ROW,COL)=STR$(NUMBER):ELSE WORKSHEET$(ROW,COL)=STRNG$
1490     NEXT COL
1500   NEXT ROW
1510   RETURN
1540   REM
1550   REM
1560   REM *****
1570   REM *   PRINT THE WORKSHEET   *
1580   REM *****
1590   REM
1600   IF BYROWS THEN WDTN=TUPLES:DEPTH=VECTORS:ELSE WDTN=VECTORS:DEPTH=TUPLES
1610   FOR ROW=1 TO DEPTH
1620     FOR COL=1 TO WDTN
1630       IF HARDCOPY THEN PRINT#2;WORKSHEET$(ROW,COL); TAB(COL*COLUMNWIDTH);:ELSE PRINT WORKSHEET$(ROW,COL);
TAB(COL*COLUMNWIDTH);
1640       NEXT COL
1650       IF HARDCOPY THEN PRINT#2;" ":ELSE PRINT
1660       NEXT ROW
1670     RETURN
1730     REM
1740     REM
1750     REM *****
1760     REM *   END-OF-PROGRAM   *
1770     REM *   CLEANUP       *
1780     REM *****
1790     REM
1800     CLOSE#1:IF HARDCOPY THEN CLOSE#2
1810     RETURN
1940     REM
1950     REM
1960     REM *****
1970     REM *   ERROR IN FILE   *
1980     REM *****
1990     REM
2000     PRINT
2010     PRINT"ERROR IN FILE..."
2020     PRINT TAB(5);"TYPE = ";TYPE
2030     PRINT TAB(5);"NUMBER = ";NUMBER
2040     PRINT TAB(5);"STRING = ";STRNG$
2050     FILEBAD=TRUE
2060     RETURN

```

## CREATING A DIF™ FILE

The following program prompts for worksheet entries (by row-column coordinate), then writes the entries on a diskette in a DIF file. Either a string or number (integer or real) can be entered. To enter a label that starts with a number, type a quotation mark (") as the first character of the label. To end a row, type **ESC-D RETURN**; to end the worksheet, type **ESC-D ESC-D RETURN**.

The program assumes the coordinate of the lower-right corner of the worksheet is the row-column coordinate of the location immediately to the left of the coordinate where **ESC-D ESC-D** is typed, so the last row should be at least as wide as all preceding rows. The worksheet is saved by rows.

```

60 REM *****
70 REM *   INITIALIZATION   *
80 REM *****
90 REM
100 esc$=CHR$(27)
110 lastinrow$=esc$
120 lastonsheet$=esc$+esc$
130 quote$=CHR$(34)
140 nul$=quote$+quote$
150 row=1
160 col=1
170 false=0
180 true=-1
190 leftarrow$=CHR$(8)
200 cr$=CHR$(13)

```

```

400 DIM worksheet$(50,50)
440 REM
450 REM
460 REM *****
470 REM *   MAIN ROUTINE   *
480 REM *****
490 REM
500 GOSUB 1000:REM          * prompt for orders
510 GOSUB 1200:REM          * prompt for entries
520 GOSUB 1400:REM          * write file
530 GOSUB 2000:REM          * end of program cleanup
540 END
940 REM
950 REM
960 REM *****
970 REM * PROMPT FOR ORDERS *
980 REM *****
990 REM
1000 HOME
1010 INPUT"Write the file (Y or N): ",reply$
1020 IF reply$="Y" OR reply$="y" THEN diskcopy=true
1030 IF NOT diskcopy THEN RETURN
1040 INPUT"File name: ",filename$
1050 IF RIGHT$(filename$,4)<>".DIF" THEN filename$=filename$+".DIF"
1060 RETURN
1140 REM
1150 REM
1160 REM *****
1170 REM * PROMPT FOR ENTRIES *
1180 REM *****
1190 REM
1200 HOME
1210 worksheet$(row,col)="
1220 PRINT"Row";ROW;" , Column ";CHR$(64+col);": ";
1230 GET reply$
1232 IF reply$<>leftarrow$ THEN 1240
1233 IF LEN(worksheet$(row,col))=0 THEN 1230
1234 PRINT leftarrow$;" ";leftarrow$;
1236 IF LEN(worksheet$(row,col))=1 THEN worksheet$(row,col)="":GOTO 1230
1238 worksheet$(row,col)=LEFT$(worksheet$(row,col),LEN(worksheet$(row,col))-1)
1239 GOTO 1230
1240 IF reply$<>cr$ THEN PRINT reply$;worksheet$(row,col)=worksheet$(row,col)+reply$:GOTO 1230
1250 PRINT
1260 IF worksheet$(row,col)=lastonsheet$ THEN width=col-1:depth=row:RETURN
1270 IF worksheet$(row,col)=lastinrow$ THEN row=row+1:col=1:PRINT:GOTO 1210
1280 col=col+1
1290 GOTO 1210
1310 REM
1320 REM
1330 REM *****
1340 REM *   WRITE FILE   *
1350 REM *****
1360 REM
1370 REM -----
1380 REM          HEADER
1390 REM -----
1400 IF NOT diskcopy THEN RETURN
1404 OPEN#1 AS OUTPUT,filename$
1410 PRINT#1;"TABLE"
1420 PRINT#1;0;" ";1
1430 PRINT#1;nul$
1440 PRINT#1;"VECTORS"
1450 PRINT#1;0;" ";depth
1460 PRINT#1;nul$
1470 PRINT#1;"TUPLES"
1480 PRINT#1;0;" ";width
1490 PRINT#1;nul$
1500 PRINT#1;"DATA"
1510 PRINT#1;0;" ";0
1520 PRINT#1;nul$
1540 REM
1550 REM -----
1560 REM          DATA RECORDS
1570 REM -----
1580 FOR col=1 TO width
1590   PRINT#1;-1;" ";0
1600   PRINT#1;"BOT"
1610   FOR row=1 TO depth
1620     IF VAL(worksheet$(row,col))>0 OR LEFT$(worksheet$(row,col),1)="-" OR LEFT$(worksheet$(row,col),1)="0" THEN
PRINT#1;0;" ";VAL(worksheet$(row,col)):PRINT#1;"V":GOTO 1660
1630     IF LEFT$(worksheet$(row,col),1)=quote$ THEN worksheet$(row,col)=MID$(worksheet$(row,col),2)
1640     PRINT#1;1;" ";0
1650     PRINT#1;quote$;worksheet$(row,col);quote$
1660     NEXT row
1670   NEXT col
1680 REM
1690 REM -----
1700 REM          END-OF-DATA
1710 REM -----
1720 PRINT#1;-1;" ";0
1730 PRINT#1;"EOD"
1740 RETURN
1930 REM
1940 REM
1950 REM *****
1960 REM *   END-OF-PROGRAM   *
1970 REM *   CLEANUP   *
1980 REM *****
1990 REM
2000 CLOSE#1
2010 RETURN

```



**VISICALC® Advanced Version****FORMULAS USED TO CALCULATE THE FINANCIAL FUNCTIONS**

Financial functions perform financial calculations, such as Present Value, Future Value, and Internal Rate of Return.

The Financial functions are presented in the table below (arguments are separated by commas).

<b>Function</b>	<b>Result</b>
@RATE(n,pmt,pv,fv)	Interest or discount rate of the given arguments.
@PERIODS(i,pmt,pv,fv)	Number of periods of the given arguments.
@PMT(i,n,pv,fv)	Amount of each payment of the given arguments.
@PV(i,n,pmt,fv)	Present Value of an <i>even</i> series of payments.
@FV(i,n,pmt,pv)	Future Value of an <i>even</i> series of payments.
@IRR( <i>investment,range</i> )	Internal Rate of Return based on an initial <i>investment</i> and a <i>range</i> of even or uneven cash flows. Both arguments must be supplied.
@NPV(i, <i>range</i> )	Net Present Value of a series of even or uneven payments; this series is the <i>range</i> argument.

The formulas used in the financial functions are listed in Appendix C.

For the purposes of Financial functions, money paid out (payments or investments) has a negative value and all money received (loans or investment income) has a positive value. Negative interest rates indicate depreciation.

Values returned by the financial functions depend on the formulas used in the functions. In certain cases, a financial function must choose between two solutions of its formula. This might cause values on your worksheet to differ from values returned by a calculator or annuity table. In such a case, you might want to know the formula used to calculate a function. For this reason, the financial functions are listed below.

**NOTE:** Abbreviations for the arguments used by the Financial functions are listed below.

- n Number of periods; must be an integer greater than zero.
- pmt Payment each period; can be positive or negative.
- i Interest rate; must be positive or negative but not zero.
- pv Present value of a series of payments. May be positive or negative but not a range.
- fv Future value of a series of payments. May be positive or negative but not a range.

### THE RATE FUNCTION-@RATE

When payment (*pmt*) is not equal to zero, @RATE is solved for *i* using an iterative method:

$$0 = pv * (i + 1)^u + pmt * (1 - (i + 1)^{-v}) / i + fv * (i + 1)^{-v}$$

where *u* is the fractional part of *n* and *v* is the integer part of *n*.

If *pmt* is 0, the following formula is used:

$$i = (fv / -pv)^{(1/n)} - 1$$

### THE PERIODS FUNCTION-@PERIODS

$$n = \ln ((fv * i) - pmt) / (-pv * i - pmt) / \ln (i + 1)$$

### THE PAYMENT FUNCTION-@PMT

$$pmt = fv * i / (1 - (i + 1)^{-n}) + pv * i * (i + 1)^n / (1 - (i + 1)^{-n})$$

### THE PRESENT VALUE FUNCTION-@PV

$$pv = (-pmt * (1 - (i + 1)^{-n}) / i + fv / (i + 1)^n)$$

### **THE FUTURE VALUE FUNCTION-@FV**

$$fv = pmt * (1 - (i + 1)^n) / i - pv * (i + 1)^n$$

### **NET PRESENT VALUE FUNCTION-@NPV**

@NPV takes two arguments: interest or discount rate (*i*) and a range of cash flows (*cf1*, *cf2*, *cf3...cfk*).

$$npv = cf1 / (1 + i)^1 + cf2 / (1 + i)^2 + cf3 / (1 + i)^3 \\ \dots + cfn / (1 + i)^n$$

where *cf1* is the first cash flow in the range and *cfn* is the last cash flow in the range.

### **THE INTERNAL RATE OF RETURN FUNCTION-@IRR**

@IRR takes two arguments: value at time 0 (*c*) and a range of cash flows (*cf1*, *cf2*, *cf3...cfn*).

The value at the beginning (*c*) is the initial investment and is typically a negative value.

This function is solved for *i* using an iterative method:

$$0 = cf1 / (1 + i)^1 + cf2 / (1 + i)^2 + cf3 / (1 + i)^3 \dots + cfn / (1 + i)^n + c$$

where *cf1* is the first cash flow in the range and *cfn* is the last cash flow in the range.

### **CALCULATING @FV AND @PMT WITH PARTIAL PERIODS**

Some situations require calculating payment or future value for a fraction of a period. For example, if a loan is taken out on the 22nd of the month, and the payment is due the first day of the month, the first period is 8 days instead of a full month.

The @PMT and @FV functions should not be used with non-integer periods. Instead, you should calculate the future value of the loan on the first payment date (the first of the next month); then calculate payment or future value using the previous future value as the new present value.

For example, assume you have a loan for \$5000, over 5 years, at 12% annual interest rate. Payments for the loan are due on the first of the month. The loan is taken out on the 22nd of the month. Making calculations on annual terms (360 days for simplicity), the first partial period is .02222222 years.

FV = Future value of a sum of dollars  
PV = Present value of a sum of dollars  
i = interest rate for one period  
n = number of periods  
 $FV = PV(1 + i)^n$

To calculate the future value for \$5000 dollars after 5 years and 8 days:

Step 1: calculate the future value at 8 days. (Interest is not compounded for the first 8 days).

$FV = 5000$  plus the interest due for 8 days  
 $= 5000 + (((.12/360)*8)*5000)$   
 $= 5000 + 13.33333\dots$   
 $= 5013.33$

Step 2: calculate the future value 5 years later using the future value from step 1 as the present value for this calculation.

$FV = 5013.33(1 + .12)^5$   
 $= 8835.21$  (rounded display)

**VISICALC® Advanced Version**

Entries are capitalized as they appear in text. When more than one page is referenced, the Primary reference is listed first.

**Special Characters**

- ! (exclamation point)
  - attribute re-display, 2-68, 3-9
  - evaluate formula, 3-144
  - recalculate worksheet, 2-95, 2-118, 3-147
- % (percent sign), 3-144
- > (greater than sign)
  - GoTo command, 2-7, 3-70
  - arithmetic symbol, 2-58, 3-59
- < (less than sign), 3-59
- " (quotation mark), 2-12, 3-89
- # (crosshatch)
  - displaying formula's value on edit line, 3-144
  - /PS# printer option, 3-98
- \$ (dollar sign), 3-18
- ( (left parenthesis)
  - options of replicate command, 3-122
- ) (right parenthesis)
  - options of the replicate command, 3-122
- ; (semicolon), 2-49, 2-149 to 2-151
- ? (question mark), 1-2, 2-8, 3-70 to 3-71
- @ (at sign), 2-31, 3-42
- / (slash)
  - commands, 2-4, 3-3
  - division operator, 2-15
- \* (asterisk)
  - multiplication operator, 2-10
  - graph format, 2-117 to 2-118, 3-39
- + (plus sign), 2-15
  - displaying, 3-7, 3-17
- (minus sign), 2-15
  - displaying, 3-7, 3-17
- . (period or decimal point), 2-12
  - displaying, 3-7, 3-17
- : (colon), 3-3, 3-106, 3-108
- ^ (carat), 2-79 to 2-80, 3-77, 3-79
- CR (credit symbol), 3-7, 3-17
- DR (debit symbol), 3-7, 3-17
- @ABS, 3-44
- @ACOS, 3-61
- @AND, 3-60
- @ASIN, 3-61
- @ATAN, 3-61
- @AVERAGE, 3-46
- @CHOOSE, 3-47
- @COS, 3-61
- @COUNT, 3-46
- @DAY, 3-52
- @DOTPROD, 3-45
- @ERROR, 3-47
- @EXP, 3-44
- @FALSE, 3-47, 3-60 to 3-61
- @FV, 2-104, 3-55, 3-58
- @HMS, 3-53
- @HOUR, 3-54
- @IF, 3-60 to 3-61
- @INT, 2-114, 3-44
- @IRR, 2-107, 3-55, 3-58 to 3-59
- @ISERROR, 3-61
- @ISNA, 3-61
- @LCHOOSE, 3-48
- @LN, 3-44
- @LOG, 3-44
- @LOOKUP, 2-59 to 2-60, 3-48
- @MAX, 3-46
- @MDY, 2-105, 3-51 to 3-52
- @MIN, 3-46
- @MINUTE, 3-54
- @MOD, 2-114, 3-45
- @MONTH, 3-53
- @NA, 3-47

@NOT, 3-60  
 @NPV, 3-55, 3-59  
 @OR, 3-60  
 @PERIODS, 3-55, 3-57  
 @PI, 3-47  
 @PMT, 2-103, 3-55, 3-57  
 @PV, 3-55, 3-58  
     series of equal payments,  
         2-102  
     series of unequal payments,  
         2-106  
 @RATE, 3-55, 3-56  
 @SECOND, 3-54  
 @SIN, 3-61  
 @SQRT, 3-44  
 @SUM, 2-31, 2-35, 3-46  
 @TAN, 2-61  
 @TRUE, 3-47, 3-60 to 3-61  
 @VMDY, 3-51 to 3-52  
 @YEAR, 3-53

## A

@LABEL and @VALUE  
     Functions, The, 3-49  
 A Pause in a Sequence  
     (figure), 3-84  
 A Reference After Deleting a  
     Column (figure), 3-26  
 A Reference Before Deleting a  
     Column (figure), 3-25  
 About the Program, 1-3  
 About This Manual, 1-1  
 absolute dates (calendar  
     functions), 3-51  
 Absolute Value  
     function—@ABS, 3-44  
 accounting  
     display attributes, 3-17 to  
         3-18  
     textbooks, 3-54  
 Adding a Series of Values,  
     2-31  
 alignment, 3-7, 3-16  
 And function—@AND, 3-60

Appendix A: Operating the  
     Apple III, A-1  
 Appendix B: Configurations  
     for the VisiCalc Program,  
     B-1  
 Appendix C: Exchanging Files  
     with DIF™ (Data  
     Interchange Format), C-1  
 Appendix D: Formulas Used to  
     Calculate the Financial  
     Functions, D-1  
 Apple III Keyboard (figure),  
     A-2  
 Arccosine function—@ACOS,  
     3-61  
 Arcsine function—@ASIN,  
     3-61  
 Arctangent function—@ATAN,  
     3-61  
 Arguments, 3-43  
 Arithmetic Functions, 3-44  
 Arithmetic Functions That Use  
     a List of Arguments, 3-46  
 Arithmetic Functions Without  
     Arguments, 3-47  
*arithmetic operator*: A symbol  
     that represents one of the  
     calculations possible with  
     the VisiCalc program:  
     + (addition), - (subtrac-  
     tion), \* (multiplication),  
     / (division), and  
     ^ (exponentiation).  
*argument*: A value or label  
     that a function needs to  
     calculate its own value.  
 Assigning Complex Formats  
     Easily - /F =  
*attribute*: A cell characteristic;  
     more than one attribute is  
     allowed in a cell.  
 Attribute Command - /A, 3-5  
 Attribute Settings (figure), 3-7

## INDEX

Attributes – /GA, 3-63  
 automatic recalculation, 3-69  
   recalculate worksheet, 2-95,  
   2-118, 3-147  
 Automatic Recalculation, An  
   (figure), 2-11  
 Average function—  
   @AVERAGE, 3-46

**B**

---

*backup*: To duplicate on a  
 second diskette the data  
 stored on a first diskette to  
 protect against possible  
 data loss on the first  
 diskette.  
 backup copies of diskettes,  
 2-19,  
 back TAB, 2-70, 3-15  
 base date (calendar functions),  
 3-51  
 Blank Command – /B, 2-13,  
 3-20  
 blank entry, 3-47  
*block*: A rectangular set of  
 cells.  
 block, replicating a, 2-91,  
 3-108, 3-115 to 3-116  
 Both Option – /TB, 3-139  
 Branch Office Budget (figure),  
 2-20  
 budget application, 2-37  
 Building a Worksheet Fast with  
 Keystroke Memory  
 Sequences, 2-91

**C**


---

Calculating a Formula (figure),  
 2-13  
 Calculating an Average, 2-85  
 Calculating Values on the Edit  
 Line, 3-143

Calendar Functions, 3-50  
*cell*: The unit on the worksheet  
 into which you can enter  
 values or labels; a cell is  
 identified by its column-  
 row coordinate, e.g., A1.  
 cell  
   alignment, 3-7, 3-16  
   blanking, 2-13, 3-20  
   displays, 2-35, 3-5, 3-33  
   formats, 2-35 to 2-36, 3-33 to  
   3-41  
   gutters, 3-7, 3-16 to 3-18  
   hiding, 2-68, 3-13  
   protecting, 2-13, 3-13 to 3-14  
   setting TAB stops in, 2-69,  
   3-15  
*cell contents*: The data in a  
 cell; a label or value.  
*cell attributes*: Cell  
 characteristics that deter-  
 mine how the cell is dis-  
 played, what kind of data  
 can be entered in it,  
 whether it is hidden, and  
 more.  
 Cells (Locations), 3-125  
 centering labels, 3-16  
 Changing an Entry on the  
 Worksheet, 2-65  
 Changing the Width of All  
 Columns, 2-24  
 Changing the Width of One  
 Column, 2-23  
 Changing Printer-Setting  
 Options – /PS, 3-98  
 characteristics, cell, 3-5 to  
 3-19  
 Choose function—  
   @CHOOSE, 3-47  
 circular references, 3-145  
 Clear All Option – /CA, 2-21,  
 3-21 to 3-22  
 Clear Command – /C, 3-21

- Clear Option – /KC, 3-84  
Clear Sheet Option – /CS, 3-21 to 3-22
- column  
  changing width of all, 2-24, 3-63  
  changing width of one, 2-23, 3-65  
  deleting, 2-55, 3-23 to 3-26  
  inserting, 2-54, 3-72 to 3-73, 3-75  
  moving, 2-92, 3-90 to 3-92, 3-95  
  replicating, 2-26 to 2-29, 3-106 to 3-122
- Column Option – /IC, 3-75  
Column Width – /GC, 3-63  
Columns Too Narrow for Values (figure), 2-25  
comma (,), 2-31, 2-35, 3-43  
Command Structure Chart, 3-3 to 3-4  
*command*: An instruction to the VisiCalc program. You begin a command by typing slash (/).
- commands  
  Attribute, 3-5  
  Blank, 3-20  
  Clear, 3-21  
  Delete, 3-23  
  Edit, 3-27  
  Format, 3-33  
  Global, 3-62  
  Go To, 3-70  
  Help, 3-70  
  Insert, 3-72  
  Keystroke Memory, 3-76  
  Move, 3-90  
  Print, 3-96  
  Repeating Label, 3-104  
  Replicate, 3-106  
  Storage, 3-127  
  Titles, 3-138  
  Version, 3-148  
  Window, 3-149
- Comparing the Format and Attribute Commands (figure), 3-6  
comparison operators, 2-57, 3-59  
complex formats, assigning, 3-40  
condition statements, 2-57, 3-59  
consolidating worksheets, 2-84, 2-87  
CONTROL keys, 2-5, A-2  
  CONTROL-C, 2-5, 2-12, 2-42  
  CONTROL-E, 3-27, 3-87  
  CONTROL-K, 3-76 to 3-80  
  CONTROL-Q, 3-88  
Controlling the Format of Entries, 2-35  
Controlling the Type of Data That Can Be Entered, 2-71  
Converting Calendar Dates to Absolute Dates—@VMDY and @MDY, 3-51  
*coordinate*: The intersection of a column and a row on the worksheet; identified by its column letter and row number.  
Copying a Single Cell, 3-109  
Copying Columns With the Replicate Command, 2-61  
Correcting Mistakes, 2-8  
Correcting Typing Errors, 2-5  
Correcting With the ESCAPE Key, 2-8  
Cosine function—@COS, 3-61  
Cost-Estimating Worksheet, A, 2-111  
Count function—@COUNT, 3-46  
Creating a Budget Calculator, 2-37  
Creating a Column of Entries from One Entry, 3-111  
Creating a DIF File, C-8

## INDEX

- Creating a Print File – /PF, 3-102
- Creating Subdirectories: The /SC Command, 3-135
- credit symbol (CR), 3-7, 3-17
- Cursor, The, 2-5, 3-125
- cursor*: A white rectangle on the screen that indicates the cell where you can enter values and labels on the worksheet.
- cumulative worksheets, 2-87
- D**
- 
- Data Area To Be Saved as a Separate File (figure), 2-77
- Data Records (of DIF file), C-5
- data storage diskette*: A diskette that has been formatted or initialized so that it can store worksheets.
- Day Function—@DAY, 3-52
- debit symbol (DR), 3-7, 3-17
- decimal places
- displaying, 3-7, 3-17
  - rounding to, 3-44
  - separating with commas, 3-7, 3-18
- default*: The setting that the program has when it is loaded, also known as the startup setting. You can change this setting or revert to the startup setting.
- decimal point, 2-12, 3-7, 3-18
- Default Format – /FD, 3-35
- Default Option – /AD, The, 3-19
- Define Sequence Option – /K = or CONTROL-K =, 3-78
- Defining a Keystroke Sequence (figure), 3-77
- Defining a Keystroke Sequence Usable Anywhere on the Worksheet, 2-89
- Defining Your Own Commands, 2-78
- Delete Command – /D, 3-23
- Delete Option – /SD, 3-131
- Deleting a File, 2-81
- Deleting Attributes, 3-10
- Deleting Columns (figure), 3-24
- Deleting Columns and Rows, 2-55
- Describing a Comparison Condition, 2-57
- device drivers, B-1 to B-2, B-4
- device names, B-1 to B-3
- DIF*: Abbreviation for Data Interchange Format. A format for storing data in a diskette file that permits data to be exchanged among VisiCalc worksheets or other programs, such as VisiSchedule™.
- DIF file
- creating, 2-76, 2-80, 3-132, C-2
  - dumping, C-6
  - format, C-4
  - printing a worksheet from, C-7
- DIF Format, The, C-4
- data records, C-5
  - end-of-data records, C-5
  - header, C-5
- DIF™ Options – S#S and S#L, 3-132
- discount rate (financial functions), 2-99, 3-55, 3-56
- disk drives, 2-16, B-1
- disk full error, 3-130
- diskette
- preparing for data storage, 2-1, 2-16, A-3
  - handling, A-2

- Diskette Files, 3-102
- display characteristics,  
controlling, 2-44 to 2-46,  
3-5 to 3-21
- displaying  
accounting symbols, 3-7,  
3-17 to 3-18  
asterisks (graph format),  
2-117 to 2-118, 3-18,  
3-39  
dollar sign (\$), 3-7, 3-17  
labels, 3-16, 3-33 to 3-40  
parentheses, 3-7, 3-17  
percent sign (%), 3-7, 3-17  
values, 3-7, 3-17
- Displaying File Names, 2-54
- Displaying the Formulas of the  
Worksheet, 2-72
- Dollars-and-Cents Format -  
/F\$, 3-38
- dollar sign (\$), 3-7, 3-17
- Dot Product function—  
@DOTPROD, 3-45
- Drawing Bar Graphs, 2-117
- Drawing Lines on the  
Worksheet, 2-32
- drivers  
device, B-1 to B-4  
printer, B-1 to B-3
- Dumping a DIF™ File, C-6
- E**
- 
- edit*: To modify a worksheet  
entry.
- Edit Command - /E, 3-27
- edit cue*: A small white rect-  
angle that marks where  
changes can be made on  
the edit line.
- edit line*: The line just above  
the column letters that dis-  
plays what is typed on the  
worksheet, or items that  
can be modified.
- Edit Line, 3-124
- Edit Option - /KE, 3-84
- Editing a File Name, 2-66
- Editing a Sequence Definition,  
2-89
- Electronic Worksheet, The  
(figure), 1-4
- Empty Worksheet (figure), 2-3
- End-of-Data Records (of DIF  
file), C-5
- Entering Labels as Titles, 2-21
- Entering a Multi-Column  
Label, 2-42
- entry*: A label or value written  
on the worksheet.
- entry line*: The top line of the  
screen that describes the  
contents of the cursor cell.
- Entry Line, 3-124
- Entry Line of Invoice (figure),  
2-48
- EOD message, 3-129, C-5
- Erasing an Entry, 2-13
- Error function—@ERROR, 3-47
- ERROR messages  
disk full, 3-130  
I/O, see 3-127, A-2, B-1  
to B-4  
Volume not found, see  
3-127, B-1 to B-4
- Examples of Maximum Func-  
tion (figure), 2-56
- Expanded Budget (figure),  
2-34
- Exponential Value Function  
—@EXP, 3-44
- expression format, 2-72, 3-10
- Expression Format Option  
- /AE, 2-72, 3-10
- expressions  
displaying on screen, 2-72,  
3-10  
printing, 2-72 to 2-73, 3-11  
to 3-12

**F**

False function—@FALSE,  
3-47, 3-60 to 3-61

*file*: A collection of data stored on a diskette. As it pertains to the VisiCalc program, it is either a worksheet or a data file stored in the DIF format.

*file name*: The name of a file stored on a diskette.

File Names, 2-17, 3-29

finance textbooks, 3-54

Financial Calculator (figure), 2-100

Financial Functions, 3-54

Financial Functions for Equal Payments, 2-106

Finding the Future Value of a Fixed Amount, 2-104

Finding the Future Value of a Series of Payments, 2-100

Finding the Interest Between Two Dates, 2-105

Finding the Internal Rate of Return, 2-107

Finding the Payment of a Loan, 2-103

Finding the Present Value of a Series of Payments, 2-102

Finding the Present Value of Unequal Payments, 2-106

First Cost-Estimating Worksheet (figure), 2-112

First Cost-Estimating Worksheet (photo), 2-113

*format*: A cell display characteristic; only one format is allowed in a cell.

*format a diskette*: Prepare a diskette to receive data for storage.

Format Command — /F, 3-33

Format Option — /GF, 3-69

Format to Attribute  
Conversions, 3-41

formats  
default, 3-35  
dollars-and-cents, 3-38  
graph, 2-117 to 2-118, 3-39  
integer, 3-37  
left-aligned, 2-36, 3-37  
repeating, 3-39  
right-aligned, 2-36, 3-38  
user-defined, 3-40

formatting diskettes, 2-1, 2-16, A-3

Formatting Integers and Dollars-and-Cents, 2-46

Formatting Left and Right, 2-44

*formula*: A mathematical statement that calculates a number and can consist of numbers, arithmetic operators, coordinates, or functions.

Formula Replicated No Change (figure), 3-118

Formula Replicated Relative (figure), 3-119

Formulas and Recalculations, 2-10

Formulas Used to Calculate the Financial Functions, D-1 to D-3

Formulas, VisiCalc, 2-15

formulas  
displaying on screen, 2-72, 3-10 to 3-12  
printing, 2-72 to 2-73, 3-10 to 3-12, 3-103

Forward and Circular  
References, 3-145 to 3-147

Forward References by  
Column and Row (figure), 3-146

*function*: A special VisiCalc statement that begins with an @ sign, uses arguments, and performs a common or complex calculation.

#### Functions

- Absolute Value (@ABS), 3-44
- And (@AND), 3-60
- Arccosine (@ACOS), 3-61
- Arcsine (@ASIN), 3-61
- Arctangent (@ATAN), 3-61
- Average (@AVERAGE), 3-46
- Choose (@CHOOSE), 3-47
- Cosine (@COS), 3-61
- Count (@COUNT), 3-46
- Day (@DAY), 3-52
- Dot Product (@DOTPROD), 3-45
- Error (@ERROR), 3-47
- Exponential Value (@EXP), 3-44
- False (@FALSE), 3-47, 3-60 to 3-61
- Future Value (@FV), 2-104, 3-55, 3-58
- Hour (@HOUR), 3-54
- Hours-Minutes-Seconds (@HMS), 3-53
- If (@IF), 3-60 to 3-61
- Integer Part (@INT), 2-114, 3-44
- Internal Rate of Return (@IRR), 2-107, 3-55, 3-58 to 3-59
- Is Error (@ISERROR), 3-61
- Is Not Available (@ISNA), 3-61
- Logarithm Base 10 (@LOG), 3-44
- Lookup (@LOOKUP), 2-59 to 2-60, 3-48
- Maximum (@MAX), 3-46
- Minimum (@MIN), 3-46
- Minute (@MINUTE), 3-54
- Modulo Arithmetic (@MOD), 2-114, 3-45
- Month (@MONTH), 3-53
- Month-Day-Year (@MDY), 2-105, 3-51 to 3-52
- Natural Logarithm (@LN), 3-44
- Net Present Value (@NPV), 3-55
- Not (@NOT), 3-60
- Not Available (@NA), 3-47
- Or (@OR), 3-60
- Payment (@PMT), 2-103, 3-55, 3-57
- Periods (@PERIODS), 3-55, 3-57
- Pi (@PI), 3-47
- Present Value (@PV), 3-55, 3-58
- Rate (@RATE), 3-55, 3-56
- Second (@SECOND), 3-54
- Sine (@SIN), 3-61
- Square Root (@SQRT), 3-44
- Sum (@SUM), 2-31, 2-35, 3-46
- Tangent (@TAN), 2-61
- True (@TRUE), 3-47, 3-60 to 3-61
- Valid Month-Day-Year (@VMDY), 3-51 to 3-52
- Year (@YEAR), 3-53
- Functions That Select Alternative Values and Labels, 3-47
- Future Value – @FV, 2-104, 3-55, 3-58
- Future Value and Present Value Calculations (figure), 2-103

**G**

- General Format — /FG, 3-36  
*global*: A command, format, or attribute setting that applies to all cells that do not have a local setting.  
 global  
   attribute settings, 3-8, 3-63  
   formats, 3-8, 3-69  
 Global Command — /G, 3-62  
 Go To Command, 3-70  
 Going Directly to a Cell, 2-7  
 Graph Format — /F\*, 3-39  
 Graph of Monthly Sales (figure), 2-118  
*gutter*: The character spaces between the right or left edge of a cell and the label or value displayed.  
 Gutters and Alignment in Cell Displays (figure), 3-16  
 gutters, 3-16, 3-17 to 3-18

**H**

- Handling Diskettes, A-2  
 hard disks, B-1 to B-4  
 header record (of DIF file), C-5  
 Help Command, The, 2-8, 3-70  
 Help Facility Options, 3-71  
 Help Facility, The, 2-8, 3-70  
 Hide Option — /AH, 3-13  
 Hiding an Entry, 2-68  
 Horizontal and Vertical Lookup Tables (figure), 3-49  
 Horizontal Option — /TH, 3-139  
 Horizontal Option — /WH, 3-149  
 Horizontal Windows (figure), 3-150  
 Hour Function — @HOUR, 3-54  
 Hours, Minutes, Seconds Function — @HMS, 3-53

**I**

- If function — @IF, 3-60 to 3-61  
 If Function Examples (figure), 2-58  
 Inaccurate Variance Report (figure), 2-96  
 Income as a Percentage of Sales (figure), 2-116  
 Income Variance Report (figure), 2-93  
 Income Variance Report, An, 2-93  
 Indicating Where the Rows and Columns Are to be Moved, 3-92  
*initialize*: Prepare a diskette to receive data for storage.  
 initializing a VisiCalc storage diskette, 2-1, 2-16, A-3  
 Insert Command — /I, 3-72  
 Inserting a Diskette (figure), A-3  
 Inserting Columns (figure), 3-73  
 Inserting Columns and Rows, 2-54  
 Inserting Multiple Columns, 2-59  
 Integer Arithmetic, 2-114  
 Integer Format — /FI, 3-37  
 Integer Part function — @INT, 2-114, 3-44  
 integrating worksheets (consolidating worksheets), 2-84, 2-87  
 interest rate (financial functions), 3-55, 3-56  
 Interest or Discount Rate — @RATE, 3-55, 3-56  
 Internal Rate of Return function — @IRR, 2-107, 3-55, 3-58 to 3-59  
 Introduction, 1-1  
 Invoice (figure), 2-41

Invoice With Data Area To Be Saved (figure), 2-76  
 Invoice With Lookup Table (figure), 2-53  
 Invoice With Several Items (figure), 2-50  
 Is Error function—@ISERROR, 3-61  
 Is Not Available function—@ISNA, 3-61

**J**

justification, see alignment

**K**

Keyboard, The, A-1  
 Keystroke Memory™  
*command*: A command that lets you save a series of keystrokes (called a sequence) that can be invoked by typing CONTROL-K and its single-character name.  
 Keystroke Memory Command — /K, 3-76  
 keystroke memory sequences  
   chaining, 3-78  
   clearing, 3-84  
   defining, 2-78, 2-89, 3-78  
   editing, 2-89, 3-84  
   erasing, 3-84  
   examples, 2-79, 2-80, 2-89, 2-90, 3-77  
   naming, 2-79, 3-77, 3-78  
   pauses in, 3-82  
   using, 2-78, 3-77 to 3-79

**L**

*label*: An entry that has no numeric value, usually used to identify other entries (such as a column heading).

Label Option — /AL, 3-16  
 label function — @LABEL, 3-49  
 Labels, 3-87  
 leap years in calendar functions, 3-51  
 Left-Aligned Format — /FL, 3-37  
 left alignment in cell, 2-44, 3-7, 3-17, 3-18, 3-37  
 listing files, 2-54, 3-129  
 listing sequences — /K RETURN, 3-86  
*load*: To put a program or data into computer memory.  
 Load DIF™ Option — /S#L, 2-86, 3-134, C-2  
 Load Option — /SL, 3-128  
 Loader diskette, A-3  
 Loading a Worksheet, 2-18  
 Loading the VisiCalc Program, A-3  
 Local, Global and Startup Settings (figure), 3-8  
*local*: Settings that are assigned to a cell and that override global settings.  
 local name, see file names  
   local settings, 2-36, 3-8, 3-33, 3-63, 3-69  
*location*: see cell.  
 Logarithm Base 10 function— @LOG, 3-44  
 Logic Functions, 3-59  
 logical value, 2-57, 3-47, 3-59  
 Looking Up Values in a Table, 2-59  
 Lookup function—@LOOKUP, 2-59 to 2-60, 3-48  
 Lookup Table for Discount Percent (figure), 2-63  
 Lookup Table for Retail Price (figure), 2-61

## INDEX

**M**

macro see Keystroke Memory™  
command

Making Backup Copies, 2-19

Making One Column Wider  
(figure), 2-23

Making One Copy of a  
Column, 3-112

Making Several Copies of a  
Column, 3-113

Making Several Copies of a  
Row, 3-114

Making a Worksheet More  
Flexible, 2-64

manual recalculation, 3-69

Maximum function—@MAX,  
3-46

merging worksheets  
(consolidating worksheets),  
2-84, 2-87

Minimum function—@MIN,  
3-46

Minute function—@MINUTE,  
3-54

Mode Option — /AM, 3-13

Mode Option Settings (figure),  
3-14

Modulo Arithmetic, 2-114

Modulo Arithmetic function—  
@MOD, 2-114, 3-45

Month function—@MONTH,  
3-53

Month-Day-Year function—  
@MDY, 2-105, 3-51 to 3-52

More on Labels and Values,  
2-12

More on the Sum Function,  
2-35

Move Command — /M, 3-90

Moving a Column, 3-95

Moving and Pointing the  
Cursor, 3-125

Moving Columns and Rows,  
2-92

Moving Rows, 3-92

Moving the Cursor, 2-5

Moving Two Columns (figure),  
3-95

Moving Two Rows Downward  
(figure), 3-93

Moving Two Rows Upward  
(figure), 3-94

**N**

names

- device, B-1 to B-4
- file, 2-17, 3-29
- subdirectory, 3-135
- volume, B-1 to B-4

naming files, 2-17, 3-29

naming subdirectories, 3-135

Natural Logarithm function—  
@LN, 3-44

Net Present Value function—  
@NPV, 3-55

None Option — /TN, 3-139

Not Available function—@NA,  
3-47

Not function—@NOT, 3-60

notation, scientific, 3-7, 3-18,  
3-143

Number of Periods—  
@PERIOD, 3-55, 3-57

**O**

One Window Option — /W1,  
3-153

operating system, Apple III,  
A-3

*option*: One choice of instruc-  
tion to the VisiCalc  
program.

options—see Commands

Options of the Replicate  
Command, 3-121

Or function—@OR, 3-160

Order of Calculation in  
Formulas, 3-143

- Order of Recalculation  
 Option – /GO, 3-66
- Other Screen Features, 2-4
- overlays, 3-21, B-4
- overlying worksheets  
 (consolidating worksheets),  
 2-84, 2-87
- P**
- 
- Pagination, 3-100
- parentheses  
 in functions, 3-43  
 options of replicate  
 command, 3-122
- pathnames, 3-30
- Payment function—@PMT,  
 2-103, 3-55, 3-57
- percentage sign (%), 3-7, 3-17
- Periods function—@PERIODS,  
 3-55, 3-57
- Pi function—@PI, 3-47
- Playing What-If, 1-5
- precision, rounding to, 3-44
- Precision of Displayed Values,  
 2-24,
- Precision of VisiCalc Values,  
 3-143
- Prefixes and Pathnames, 3-30
- preparing data diskettes, 2-1,  
 2-16, A-3
- Present Value function—@PV,  
 3-55, 3-58
- Print Command – /P, 3-96
- printer drivers, B-1 to B-3
- Printer-Setting Options—/PS,  
 3-98  
 Linefeed option – /PSL, 3-98  
 Page Break option – /PSB,  
 3-98  
 Titles option – /PST, 3-98  
 Page Number option – /PS#,  
 3-98
- Page Length option – /PSP,  
 3-98
- Page Width option – /PSW,  
 3-98
- Paper Length option – /PSF,  
 3-98
- Margin option – /PSM, 3-98
- Edit Setup String  
 option – /PSS, 3-98
- Replace Setup String  
 option – /PSR, 3-98
- Printer-Setting Pagination  
 (figure), 3-100
- Printing a Worksheet from a  
 DIF™ File, C-7
- Printing Formats, Attributes,  
 and Formulas, 3-103
- Printing on the Printer – /PP,  
 3-97
- Printing the Formulas of the  
 Worksheet (figure), 2-73
- Printing the Worksheet, 2-38
- Printing the Worksheet in  
 Sections, 3-101
- Printing Worksheet Formulas,  
 3-103
- ProFile™ hard disk, B-1 to B-2,  
 B-4
- Program diskette, A-3 to A-4
- Programs that Work With the  
 DIF™ Format, C-6  
 Creating a DIF File, C-8  
 Dumping a DIF File, C-6  
 Printing a Worksheet from a  
 DIF File, C-7
- prompt line*: The second line of  
 the screen; describes what  
 can be typed next.
- Prompt Line, 3-124
- Protecting Cells Against  
 Change, 2-72
- Protecting Your Work, 2-18

## INDEX

**Q**

- question mark (?)
  - Help command, 3-70
  - Putting "?" in a label, 3-88
- Quit Option - /SQ, 3-131

**R**

- range*: A series of adjacent cells in a row or column, specified by the coordinate of the first cell, an ellipsis, and the coordinate of the last cell.
- Rate function—@RATE, 3-55, 3-56
- recalculation*: The process of returning the value of each formula on the worksheet.
- Recalculation by Column (Correct Result) (figure), 3-67
- Recalculation by Row (Incorrect Result) (figure), 3-68
- Recalculation Order, 3-145
- Recalculation Priority
  - Option - /GR, 3-69
- records, DIF files, C-1 to C-8
- references, circular, 3-145
- references, forward, 3-145
- References to Other Cells, 3-141
- repeat-action on the Apple III keyboard, 2-6
- Repeating Label Command - /—, 3-104
- Repeating Labels (figure), 3-105
- Repeating Labels - /F -, 3-39
- replicate*: The process of copying an entry or range of entries to another part of the worksheet.
- Replicate Command - /R, 3-106
- Replicate Command Prompt (figure), 2-27
- replicating
  - block, 2-91, 3-115, 3-116
  - column, 3-112, 3-113
  - format, 2-36, 3-121
  - formulas, 3-117
  - rectangle, 2-91, 3-115, 3-116
  - reference to other cells, 2-29, 3-121
  - row, 3-114
- Replicating—The Key To Building a Worksheet, 2-26
- Replicating a Block Once, 3-115
- Replicating a Block Once (figure), 3-115
- Replicating a Block Several Times, 3-116
- Replicating a Block Several Times (figure), 3-116
- Replicating a Column, 2-28
- Replicating a Column Several Times (figure), 3-113
- Replicating a Format, 2-36
- Replicating a Rectangular Area, 2-91
- Replicating a Relative Reference (figure), 2-30
- Replicating a Row, 3-114
- Replicating a Row Several Times (figure), 3-114
- Replicating Attributes or Cell Contents - (A and (C, 3-122
- Replicating Cell References in Formulas: All Relative or All No Change - (N and (R, 3-121
- Replicating Formulas, 3-117
- Replicating One Cell Across a Row (figure), 3-110
- Replicating One Cell Down a Column (figure), 3-111

- Replicating References to  
Other Cells, 2-29
- Resolving Forward References,  
2-81
- reverse TAB, 2-70, 3-15
- Review Sequences Option – /K  
RETURN, 3-86
- Right-Aligned Format – /FR,  
3-38
- right alignment in cell, 3-7,  
3-17, 3-18
- Rounding for Less Precision:  
@ROUND(v, precision),  
3-44
- rounding numbers, 3-44
- Row Option (Insert) – /IR, 3-74
- rows  
deleting, 2-55, 3-23 to 3-26  
inserting, 2-54, 3-74  
moving, 2-92, 3-92  
replicating, 3-114
- S**
- Sales Report Worksheet for  
Data Files (figure), 2-84
- sample files on Program  
diskette, 1-2, 2-1, 2-67,  
2-76
- Save DIF™ Option – /S#S,  
3-132
- Save Option – /SS, 3-129
- Saving a DIF File Using a  
Keystroke Memory  
Sequence, 2-80
- Saving a Portion of the  
Worksheet, 2-76
- Saving a Worksheet, 2-17
- saving data on diskette—see  
Storage command
- Saving Your Work, 2-16
- scientific notation, 3-7, 3-18,  
3-143
- screen*: The display device of  
the computer on which the  
worksheet is viewed.
- Screen Features, Other, 2-4
- Screen Split Vertically  
(figure), 2-49
- Screen, The, 3-123
- scrolling*: The appearance of  
moving the window left,  
right, up, or down to view  
off-screen parts of the  
worksheet.
- scrolling printer-setting  
options, 3-99
- scrolling sequences – /K  
RETURN, 3-86
- Scrolling the Window, 2-5
- Second Cost-Estimating  
Worksheet (figure), 2-116
- Second function—@SECOND,  
3-54
- Selecting a Value that Meets  
a Condition, 2-56
- Selecting the Largest  
of Several Entries, 2-55
- self-defined formats, 3-40
- Sending Setup Strings, 3-100
- setting  
attributes, 3-5 to 3-21  
formats, 3-33
- Setting a Local Format, 2-36
- Setting Manual Recalculation,  
2-95
- Setting Tab Positions, 2-69
- sequence*: A series of  
keystrokes that can be  
invoked by typing its  
single-character name.  
See also Keystroke  
Memory command.

## INDEX

- sequences, Keystroke Memory  
 clearing, 2-78, 2-79, 3-84  
 chaining, 3-78  
 defining, 2-78, 2-89, 3-78  
 editing, 2-89, 3-84  
 erasing, 3-84  
 examples, 2-79, 2-80, 2-89,  
 2-90, 3-77  
 naming, 2-79, 3-77, 3-78  
 pauses in, 3-82  
 using, 2-78, 3-77 to 3-79
- setup string*: A series of  
 instructions to your  
 printer.
- Shipping Addresses of Invoice  
 (figure), 2-44
- Silentype™ printer, B-1
- Sine function—@SIN, 3-61
- Single and Double Lines on the  
 Worksheet (figure), 2-33
- Single Column Width—/GCC,  
 3-65
- SOS (operating system), A-3
- source range*: The cell or  
 range of entries to be  
 copied (replicated) to  
 another part of the  
 worksheet.
- special keys  
 in sequences, 2-79, 3-79
- Specifying the Rows or  
 Columns to be Moved,  
 3-91
- Splitting the Screen, 2-48
- Square Root function—  
 @SQRT, 3-44
- starting out  
 initializing diskettes, 2-1,  
 2-16, A-3  
 loading, A-3
- Startup Screen (figure), 3-123
- startup settings, 3-8
- Status Area, 2-4, 3-124
- status area*: The top three lines  
 of the screen, divided into  
 the entry line, prompt  
 line, and edit line.
- Storage Command—/S, 3-127
- store*: To save a worksheet on a  
 data diskette for later  
 retrieval.
- Subdirectories, 3-135, B-4
- suffixes, 3-29
- Sum function—@SUM, 2-31,  
 2-35, 3-46
- Summary Sales Report With  
 One Data Row (figure),  
 2-86
- suppressing  
 immediate recalculation,  
 2-95, 3-69  
 immediate re-display of  
 attribute, 2-69, 3-9  
 zeroes in value displays,  
 3-18
- Synchronized Scrolling  
 Option—/WS, 3-153
- 
- T**
- 
- TAB key, 2-69 to 2-70, 3-15
- Tab Option—/AT, 3-15
- tab stops, 2-69 to 2-70, 3-15
- Tangent function—@TAN, 2-61
- target range*: The cell or range  
 of cells into which the  
 other cells are copied  
 (replicated).
- template*: A structured  
 worksheet for entering  
 data.
- term (financial functions),  
 2-99, 3-55, 3-57
- textbooks, accounting and  
 finance, 3-54

- The @LABEL and @VALUE Functions, 3-49
- The Clear All Option - /CA, 3-22
- The Clear Option - /KC, 3-84
- The Clear Sheet Option - /CS, 3-22
- The Cursor, 2-5, 3-125
- The Day Function—@DAY, 3-52
- The Default Option - /AD, 3-19
- The Define Sequence Option - /K = or CONTROL-K =, 3-78
- The DIF Format, C-4  
Data Records, C-5  
End-of-Data Records, C-5  
Header, C-5
- The Dot Product Function, 3-45
- The Edit Option - /KE, 3-84
- The Electronic Worksheet (figure), 1-4
- The Expression Format Option - /AE, 2-72, 3-10
- The Functions—@, 3-42
- The Help Command, 2-8
- The Help Facility, 3-70
- The Hide Option - /AH, 3-13
- The Hour Function—@HOUR, 3-54
- The Hours, Minutes, Seconds Function—@HMS, 3-53
- The Keyboard, A-1
- The Label Option - /AL, 3-16
- The Minute Function—@MINUTE, 3-54
- The Mode Option - /AM, 3-13
- The Modulo Function, 3-45
- The Month Function—@MONTH, 3-53
- The Review Sequences Option - /K RETURN, 3-86
- The Save DIF™ Option - /S#S, 3-132
- The Screen, 3-123
- The Second Function—@SECOND, 3-54
- The Status Area, 2-4, 3-124
- The Tab Option - /AT, 3-15
- The Value Option - /AV, 3-17
- The VisiCalc Pocket Reference, 1-3
- The VisiCalc Screen (figure), A-4
- The VisiCalc Tutorial, 2-1
- The Window and Worksheet, 3-125
- The Year Function—@YEAR, 3-53
- time functions, 3-50
- Title and Column Headings of Budget (figure), 2-22
- Titles Command - /T, 3-138
- transferring data between worksheets with DIF™, 2-76, 2-80, 3-132, C-2
- Totals Block of Invoice (figure), 2-51
- Trigonometric Functions, 3-61
- True function—@TRUE, 3-47, 3-60 to 3-61
- tuple, C-4 to C-5
- tutorial files on Program diskette, 1-2, 2-1, 2-67, 2-76
- Tutorial, The VisiCalc, 2-1
- Typeahead, 3-126
- type-checking functions, 3-49

---

**U**

- user-defined formats, 3-40
- Unsynchronized Scrolling Option - /WU, 3-153
- Using a Worksheet as a Data-Entry Form, 2-82
- Using DIF Files To Create a Summary Report, 2-84

## INDEX

Using Keystroke Memory  
 Sequences for Common  
 Worksheet Features, 2-88

Using Pauses (^P) in  
 Sequences, 3-82

Using the Cursor To Refer to  
 Cells, 2-14

Using the DIF™ Format within  
 the VisiCalc Program, C-3

**V**

Valid Month-Day-Year  
 function—@VMDY, 3-51  
 to 3-52

Value Option — /AV, 3-17

value function—  
 @VALUE, 3-49

*value*: A worksheet entry used  
 in calculations; the entry  
 can be a number, a refer-  
 ence, a formula, or a  
 function.

value, logical, 2-47, 3-57, 3-59

Values, 3-140

Version Command — /V, 3-148

Vertical Option (Titles  
 command) — /TV, 3-139

Vertical Option (Window  
 command) — /WV, 3-151

Vertical Windows (figure),  
 3-152

VisiCalc Commands, 2-4, 3-3  
 to 3-4

VisiCalc Formulas, 2-15

VisiCalc Pocket Reference, 1-3

VisiCalc Screen, The (figure),  
 A-4

volume names, B-1

**W**

What You Need, A-1

width, column, 2-23, 3-24, 3-63  
 to 3-65

*window*: That portion of the  
 worksheet visible on the  
 screen.

Window and Worksheet, The,  
 3-125

Window Command — /W, 3-149

*worksheet*: A grid of rows and  
 columns on which labels  
 and values are written.

worksheet

- changing, 2-65, 3-27
- loading, 3-128
- printing, 3-96
- saving, 3-129
- writing on, 2-9

Worksheet After Inserting Row  
 (figure), 3-75

Worksheet Before Inserting  
 Row (figure), 3-74

Worksheet Displayed in  
 Expression Format  
 (figure), 3-12

Writing on the Worksheet, 2-9

**XYZ**

Year Function—@YEAR, 3-53

zeros, suppressing display with  
 /AVZ, 3-18



# READER'S CRITIQUE

**VISICALC®** USER'S GUIDE  
Advanced Version  
10950-2210

VisiCorp's Publications Department wants to provide documents that meet the needs of all our product users. Your feedback helps us produce better manuals.

Please tell us how you rate this manual in the following areas:

	Excellent	Good	Average	Fair	Poor
1. Completeness of information.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2. Clarity of information.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3. Organization of manual.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4. Appearance of manual.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5. Quantity of examples.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6. Usefulness of examples.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7. Quantity of photographs and illustrations.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8. Appropriateness of photographs and illustrations.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
9. Degree to which this manual meets your expectations and needs.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
10. Appropriateness of the manual level to a person of your experience.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
11. Overall manual.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
How much computer experience do you have?		2 years or less <input type="checkbox"/>	2-5 years <input type="checkbox"/>	Over 5 years <input type="checkbox"/>	

Please specify, by page, any errors you found in this manual.

We would appreciate hearing any other comments and suggestions you have about this manual.

Name \_\_\_\_\_

Company \_\_\_\_\_

Street \_\_\_\_\_

City \_\_\_\_\_ State \_\_\_\_\_ Zip \_\_\_\_\_



No Postage  
Necessary  
If Mailed  
in the  
United States

---

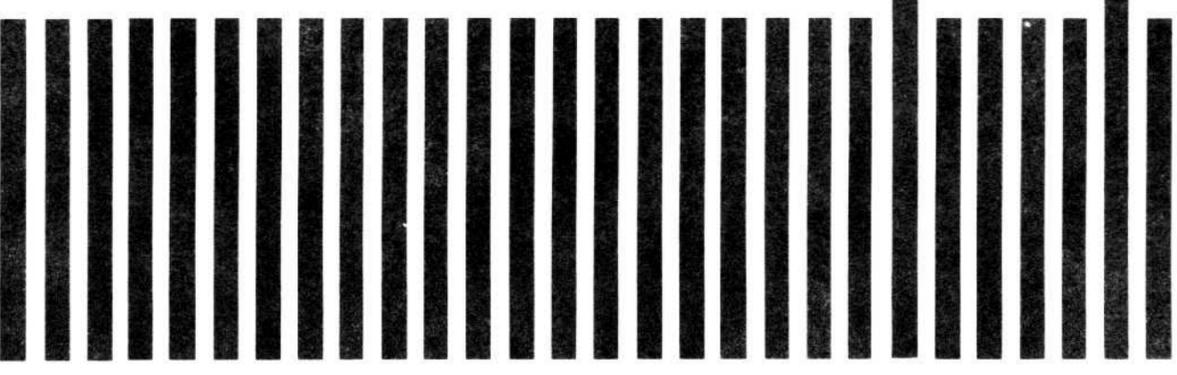
# BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 7066 SAN JOSE, CA 95134

---

POSTAGE WILL BE PAID FOR BY ADDRESSEE

VISICORP  
2895 Zanker Road  
San Jose, CA 95134







**VISICORP**<sup>™</sup>  
PERSONAL SOFTWARE<sup>™</sup>

2895 Zanker Road  
San Jose, California 95134  
Telephone 408/946-9000

Apple III  
VisiCalc Advanced Version  
10950-2210 8/82

---

# Important Information

The enclosed pages contain important new information for the User's Guide.

Please read this material *before* reading the User's Guide.

---

# Update Pages

Insert these pages in your User's Guide, as indicated below.

**Pages:**

xi to xii  
3-55 to 3-60  
D-3 to D-4

**Replace with pages:**

xi to xii  
3-55 to 3-60  
D-3 to D-4



---

# Important Information

The enclosed pages contain important information about this User's Guide.

Please read this material *before* reading the User's Guide.

---

# Update Pages

Insert these pages in your User's Guide, as indicated below.

<b>Pages:</b>	<b>Replace with pages:</b>
2-17 to 2-18	2-17 to 2-18
2-19	2-19
2-23	2-23
3-13 to 3-14	3-13 to 3-14
3-29 to 3-30	3-29 to 3-30
3-32	3-32
3-33	3-33
3-40	3-40
3-73	3-73
3-78	3-78
3-101	3-101
3-135 to 3-136	3-135 to 3-136
A-1	A-1
A-3 to A-4	A-3 to A-4
B-5	B-5
D-2	D-2



## File Names

A saved worksheet is called a *file*. The name that identifies a file is called a *file name*. Because a diskette can hold many worksheets, you must give each worksheet on the same diskette a different file name. This way, you can find the worksheet later and load it into the computer.

There are a few simple rules for creating a valid file name:

- It can be up to 15 characters long.
- It can include only letters, numbers, and periods.
- The first character must be a letter.
- It cannot include a space or RETURN.

If the storage diskette is in a drive other than drive 1 (the built-in drive), you must add a prefix to the file name that specifies the drive number. The prefix consists of a period (.), the letter D, the drive number, and a slash (/). To specify a file named BUDGET.VC in drive 2, for example, you would type .D2/BUDGET.VC.

When you specify a drive with a prefix, the VisiCalc program uses that drive for all subsequent disk operations unless you specify another. If you save a file named .D2/BUDGET.VC then later save a file named FORECAST.VC, the VisiCalc program tries to save FORECAST.VC on a diskette in drive 2. To save it on a diskette in drive 1, you must specify the drive with a prefix: .D1/FORECAST.VC.

**NOTE:** It doesn't harm anything to specify the diskette drive the VisiCalc program would use otherwise. If you save a file named .D2/BUDGET.VC and then save another a few minutes later with the name .D2/FORECAST.VC, everything works fine. Specifying the drive under these circumstances is merely redundant, not harmful.

## Saving a Worksheet

To save the worksheet, type /S. The prompt line reads Storage: L S D Q # C. Type S. The prompt line now reads Storage: File for saving, and the edit cue is on the edit line. If you have a one-drive system, open the door of the drive, remove the VisiCalc program diskette, and put in an initialized diskette. If you have a two-drive system, put the initialized diskette in drive 2.

The VisiCalc program is waiting for you to type the file name. Because this worksheet is an example, call it EXAMPLE. If you have a one-drive system, type EXAMPLE. If you have a two-drive system, type .D2/EXAMPLE to save the worksheet on the diskette in drive 2.

Now type RETURN. The disk drive begins whirring and the light comes on. After a moment the drive quiets down, the light goes off, and the prompt and edit lines clear. The worksheet is safely filed away on the storage diskette.

If you have a one-drive system, replace the VisiCalc program diskette.

## Loading a Worksheet

To reassure yourself that the example worksheet was really saved, clear the worksheet (type /**CAY**). Type /**SL** (the Load option of the Storage command). The prompt line reads Storage: File to load.

If you have a one-drive system, remove the VisiCalc program diskette and replace it with the diskette on which you saved the worksheet.

Type **EXAMPLE RETURN**. Again, the disk drive begins whirring and, in a few moments, your worksheet is back on the screen. It remains on the disk until you erase it or replace it by storing another worksheet with the same name.

If you have a one-drive system, remove the storage diskette and replace it with the VisiCalc program diskette.

## Protecting Your Work

Have you ever worked out a problem or made some notes to yourself on a sheet of paper, only to find later that you lost the sheet or that someone accidentally threw it away? If you should lose power before you save the worksheet on a diskette, all the work you have done is lost.

To protect yourself, you should periodically save the worksheet on diskette. As you work, think how long it has been since you last saved the worksheet. If you have spent more time than you would care to lose if something goes wrong, or if you have results which might be difficult to reconstruct, then it's time to save the worksheet.

Printing the worksheet is another security measure. You can print not only the worksheet, but also the formulas and formats behind it. See "Print Command" and "Storage Command" in Chapter 3 for descriptions of how to print the content and structure of the worksheet. With printed copies of the worksheet and its underlying formats and formulas, you can reconstruct the worksheet without a disk file, if necessary.

## **Making Backup Copies**

Although a diskette is a safe and reliable medium for storing information, saving your work periodically is only the first step in protecting yourself. A diskette can be scratched or pick up grease and dust. It can be damaged by heat, exposed to a magnetic field, or accidentally initialized (which erases it). Eventually it wears out. The average lifetime of a diskette is about 40 hours of use (whenever the disk drive light is on, the diskette is in use). For complete protection, you should make backup copies of your files on separate diskettes. You can make a backup copy of an entire disk with the disk copy program on your Apple III Utilities Disk.

Use the Storage command to make a backup copy of a file. Insert the diskette that contains the file you want to copy into drive 1 (or whatever drive you are currently using). Then use the Load option of the Storage command (/SL) to load the worksheet. Remove the diskette, insert an initialized diskette, and use the Save option of the Storage command (/SS) to save the same worksheet on the second (backup) diskette.

## **SUMMARY**

This completes Lesson One. You may wish to experiment for a while, moving the cursor around and writing your own labels, numbers, and formulas. Try writing some formulas by pointing the cursor to specify the coordinates. Use the Help command to refresh your memory if you forget a detail.

The VisiCalc program has many more features not yet described, and you may stumble upon some of them. If something happens that you don't understand, make a note of it, or get a description with the Help command and experiment. When you're ready to continue, turn to Lesson Two.

## LESSON TWO

Entering Labels as Titles	2-21
Changing the Width of One Column	2-23
Changing the Width of All Columns	2-24
Precision of Displayed Values	2-24
Replicating — The Key to Building a Worksheet	2-26
Replicating a Column	2-28
Replicating References to Other Cells	2-29
Adding a Series of Values	2-31
Drawing Lines on the Worksheet	2-32
More on the Sum Function	2-35
Controlling the Format of Entries	2-35
Setting a Local Format	2-36
Replicating a Format	2-36
Creating a Budget Calculator	2-37
Printing the Worksheet	2-38
Summary	2-39

In Lesson One, you learned the basic operating techniques of the VisiCalc program. In this lesson you will create a typical budget — in this instance, for a branch office. Figure 2-6 is a printed copy of the budget.

By the time you finish this lesson, you will know enough about the VisiCalc program to create some worksheets of your own.

Budget  
1983 Est.

Salaries	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Total
Secretary	1000	1000	1000	1000	1000	1100	1100	1100	1100	1100	1100	1100	12700
Receptionist	800	800	800	800	800	800	800	800	800	800	800	800	9600
Accts. Clerk	850	850	850	850	850	850	850	850	850	0	0	0	7650
Office Mgr.	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250	15000
	3900	3900	3900	3900	3900	4000	4000	4000	4000	3150	3150	3150	44950
-----													
Phys. Plant	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Total
Rent	1300	1300	1300	1300	1300	1300	1300	1500	1500	1500	1500	1500	16600
Utilities	200	200	200	200	200	200	160	160	160	160	160	160	2160
Insurance	400	400	400	400	400	400	400	400	400	400	400	400	4800
Maintenance	150	150	150	150	150	150	150	150	150	150	150	150	1800
Furniture	500	500	500	500	500	500	500	500	500	500	500	500	6000
	2550	2550	2550	2550	2550	2550	2510	2710	2710	2710	2710	2710	31360
	6450	6450	6450	6450	6450	6550	6510	6710	6710	5860	5860	5860	76310

010-002

**Figure 2-6. Branch Office Budget**

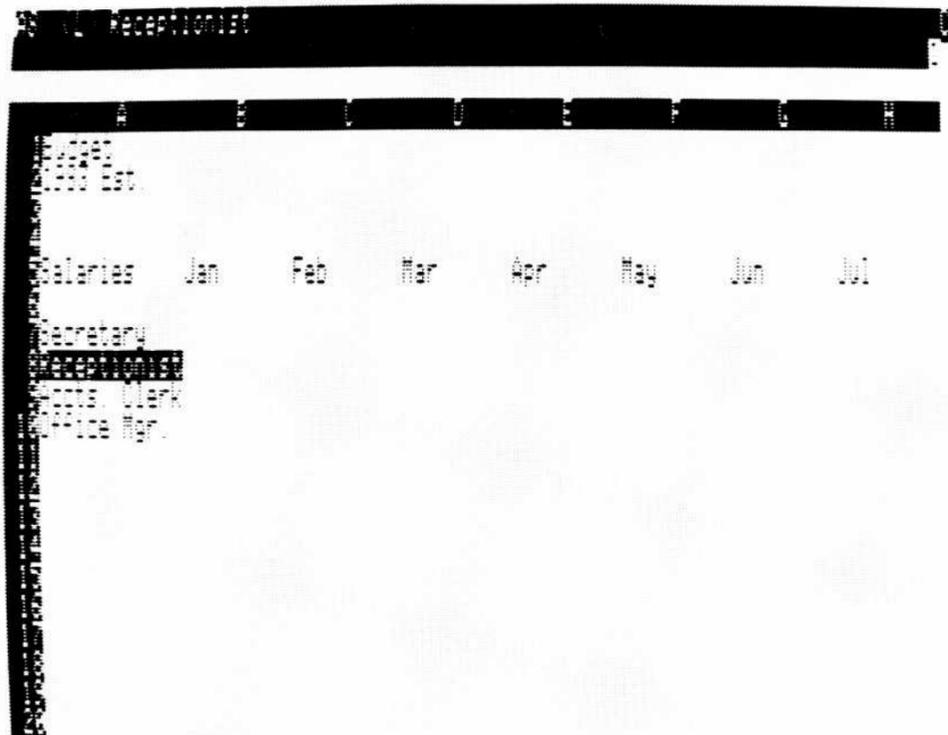
## CHANGING THE WIDTH OF ONE COLUMN

You can change the width of a column to display a wide label or value. Such a change is called a *global* change, because it affects more than a single location. Type **/G** (the Global command).

The prompt line reads: Global: A C O R F. The characters following Global: are options of the Global command. You want **C**, the column-width option. The other options are described later.

Type **C** to set the global column width. The prompt line reads Global: Column width: # or C. If you typed a number now, the width of all columns would be changed. You want to change the width of just one column, so type **C**. Now the prompt line reads Global: Column Width.

You can set individual columns to any width from 1 to 125. The longest label you entered has 12 characters, so type **12 RETURN**. Now the entire label is displayed in Column A. The screen should look like Figure 2-8.



010-007/P

Figure 2-8. Making One Column Wider

Now to enter some values. Type the following to enter the January numbers:

```
> B7 RETURN  
1000 ↓ 925 ↓ 850 ↓ 1250 RETURN
```

## CHANGING THE WIDTH OF ALL COLUMNS

Besides changing the width of one column, you can change the width of all columns. It's often helpful to make the columns narrow so that more fit on the screen. To see this, type **/GC5 RETURN**.

## PRECISION OF DISPLAYED VALUES

The VisiCalc program displays a number to the greatest precision possible in the space available. It leaves the first space of a column blank to keep columns of numbers from running into each other, so the space available to display a value is 1 less than the column width. What if there isn't room to display all the digits of a number?

Make the columns even narrower by typing **/GC4 RETURN**, and look at B7. Instead of 1000, it reads 1E3. This means "1 times 10 to the 3rd power," or 1000. Even though there isn't room for all four digits of 1000, it displays the value. But look at B10; instead of 1250 it, too, reads 1E3. That is the maximum precision possible in 4-character columns, so 1000 and 1250 are both displayed as 1E3.

What if there simply isn't room to display a number in the available space? If you specified that all these numbers were to be displayed as integers, it would require a column width of at least 5 to display them. Change the global format to Integer by typing **/GFI**. Now look at B7 and B10. They read >>>. If the VisiCalc program cannot display a value because the columns are too narrow, it fills its location with greater-than signs (>) to warn you.

### **The Hide Option—/AH**

The Hide option makes the contents of a cell invisible.

When you type /AH, Hide: Y N D! appears on the prompt line. The three option settings are:

- Y Yes, hide the contents of this cell.
- N No, don't hide the contents of this cell.
- D Set this cell to the default last set with /GAH

If you type anything other than Y, N, or D in response to the prompt, you will return to the Attribute command prompt and the cell will be left unchanged.

The /AH option overrides /AV (Value) and /AL (Label). The /AE option, which displays formulas and expressions in cells, overrides /AH.

### **The Mode Option—/AM**

The Mode option controls whether or not data can be entered in a cell, and what kind of data can be entered (labels, values or numbers).

When you type /AM, this prompt appears: MODE: P U A L V # D!. The settings for this attribute are:

- P Protects the contents of a cell from being changed; no entry is allowed in the cell.
- U Allows cell contents to be changed.
- A Allows you to enter all data types—numbers, formulas, and labels.
- L Allows you to enter only labels; numbers and formulas are not allowed.
- V Allows you to enter only numbers and formulas; labels are not allowed.
- # Allows you to enter only numbers; formulas and labels are not allowed.
- D Returns setting to global default last set with /GAM.

(The ! switches between immediate and delayed re-formatting.)

Figure 3-5 illustrates the various settings of the Mode option.

Settings	Cell will accept	Cell will not accept
/AMV (only values)	+B7/B24-A2 @SUM (A1...A10) 55.7	LABEL 1 "-GROSS-+@"
/AML (only labels)	LABEL 3 "-GROSS-+@"	+B14-A2 55.7
/AM# (only numbers)	24.5691 10	+B7/B24-A2 LABEL 3 @SUM (A1...A10)
/AMA (all)	55.7 @SUM (A1...A10) "-GROSS-+@" LABEL 1 24.5691	

010-019

**Figure 3-5. Mode Option Settings**

The Mode option can save you time entering values and labels. When you have the Mode option set to L (labels only) you do not need to begin a label whose first character is non-alphabetic with a quotation mark ("). For instance, if the label-only setting (/AML) is in effect in a cell, you can type the first character in the label -NET- without typing " first. When the first character is a command character—/, ;, >, [SPACE]—you must type the quotation mark (") first, however.

Likewise, if the Mode is set to /AMV (values only), you do not need to type + to begin a formula whose first character is a cell coordinate (A1/B1 - C1). If you point to a cell with the cursor to enter a value, however, you must type the + first.

You can change attributes and formats in a protected (/AMP) cell, and you can replicate into a protected cell; /AMP protects only data entry. Replicating a value into a cell in which /AML (labels only) has been set causes an error; this is also true for replicating a label into a cell where /AMV or /AM# is in effect.

## **FILE NAMES**

Each time you save a worksheet to a storage device, such as a diskette, you create a file. Every file on a diskette must have a unique name, known as its file name, to differentiate it from all other files on the diskette. If you do not use subdirectories (see "The Storage Command" in the Reference and below), the file name is called the local name. If you use subdirectories, you can use the same local name once per subdirectory.

If you try to save a file under a local name that you've given to an earlier file, the contents of the more recent file will take the place of the contents of the earlier file, effectively erasing it. You should save to an existing file name only when you are updating the existing file.

A valid local name or subdirectory name can include up to 15 characters. The first character must be a letter. The rest of the name can include letters, numbers, and periods. No other characters, including the space character, are allowed in a local or subdirectory name.

Thus, FILETWO.VC and FILE2.DIF are both valid. Two examples of invalid names are 1980ACCOUNTS (begins with a number) and MYDISKETTE NO.TWO/PF (includes a space, a slash, and is longer than 15 characters).

Note the suffixes in the above local name examples—.VC, .DIF, and .PF. They represent the three types of files that VisiCalc can create. Their purpose is to help you and VisiCalc differentiate between the three file types.

We recommend that you append .VC to any standard VisiCalc local name (one saved with /SS), .DIF to any DIF format local name (one saved with /S#S), and .PF to any Print Format local name (one saved with /PF). See Part III, "The Storage Command" and "The Print Command" for further details.

## PREFIXES AND PATHNAMES

Whenever you format a diskette (see Appendix A), SOS (the Apple III Operating System) not only prepares the diskette to receive data, but also creates a directory known as the main or root directory. The directory stores the file names and file addresses of all the files on its disk.

Because the directory is a file, SOS gives it a name, the volume name. The formatting process creates the name /BLANK. With the Apple III Utilities Diskette, you can change the volume name on any diskette; i.e., /BUDGET, /MYDISK1, /SALES3, etc. In fact, it's a good idea to give each volume a unique name. See the *Apple III Owner's Guide* for details on how to rename a diskette directory.

Besides naming a directory and recognizing directory names, SOS also recognizes device names. Every device connected to the Apple III has a name. For example, the built-in disk drive is .D1. The other floppy drives are .D2, .D3, and .D4. If you are using the ProFile hard disk, the device name is .PROFILE. The device name for the Corvus hard disk is .C1. If you have connected a Silentype™ printer, SOS recognizes it as .PRINTER.

Either a device name or a volume name can serve as a prefix to the local name of a file. Note that a device name always begins with a period (.) and a volume name always begins with a slash (/). The slash also serves to separate either type of prefix from the local name. The prefix-slash-local name combination is known as a pathname.

If you have no extra drive(s) and are using the Storage commands (see "The Storage Command"), you have no need for prefixes because the built-in drive and whatever volume occupies it are in the default condition. That is, SOS automatically refers to them for any Storage command operations.

However, if you want to locate a file on a drive other than the default drive, you must use the pathname. As an example, let's say that the built-in drive was the default drive and that drive 2 contained a diskette with the volume name /SALES and a local file named FORECAST.VC. To load FORECAST.VC, you would first type /SL, followed by either .D2/FORECAST.VC or /SALES/FORECAST.VC.

When you specify a device or volume with a prefix, SOS makes that prefix the default value. You won't have to specify a prefix again until you go to another device or volume.

## **SUBDIRECTORIES**

SOS also lets you create subdirectories with the /SC command (see "The Storage Command"). A subdirectory is a file that points to other files, even to other subdirectories. You don't have to store your files by subdirectories, but they are extremely handy for hierarchical structuring of data.

Once you have created a subdirectory, it becomes the default reference for all storage commands. To gain access to other files and subdirectories, you must name them as part of a pathname.

For example, let's say you have two subdirectories on a diskette, one named SUB1 and the other named SUB2. Further, each of these subdirectories contains two files, FILE1 and FILE2. If you are working with FILE1 in SUB1 and want to use FILE1 in SUB2, you would type /SL, the Load command, followed by .D1/SUB2/FILE1. This assumes that the diskette containing these files is in the built-in drive.

Note that after typing .D1/SUB2/, you can scroll through the local names in SUB2 (see "Scrolling" below).

Subdirectories cannot be deleted until you have first deleted all the files that they point to. When this is done, you must use the regular SOS DELETE command. /SD will not delete subdirectories. For a thorough discussion of subdirectories, see the *Apple III Owner's Guide* and "The Storage Command" in this chapter.

## **SCROLLING**

To find your files more quickly, VisiCalc allows you to scroll through the file names in a diskette directory. For example, if you want to load a file, scroll through the local names by typing /SL and pressing →. The drive whirs and displays the first file name it finds on the edit line.

To view other file names, press → again. You can press → until you have scrolled through every file name in that directory. You can also scroll backwards by pressing ←; however, ← works only after you have pressed →. To select a file for loading, scroll to the file name you want, and press RETURN.

If you are using subdirectories, scrolling operates a little differently. After you load the program, press → to put the names in the main diskette directory on the edit line. To enter a subdirectory, scroll to its name and press RETURN. You can then use → and ← to scroll through the subdirectory names. When you reach the last name in the subdirectory, press → to place the subdirectory name on the edit line. You can type a new file name to add to the subdirectory, or you can press → to continue scrolling through the names in the main directory.

When you enter a subdirectory, the prefix for that directory is set (see the *Apple III Owner's Manual*). The next time you scroll through file names, the names in that subdirectory appear. If you wish to enter a different directory, type a different directory prefix when you are prompted for the file name. You can also enter a different directory when a file name is on the edit line. Pressing ↑ will move the edit cue back to the previous slash, deleting the characters in between. You can press ↑ until you have deleted the subdirectory name and type in a new name, or you can press ↑ once and press → to scroll through names in the next higher directory.

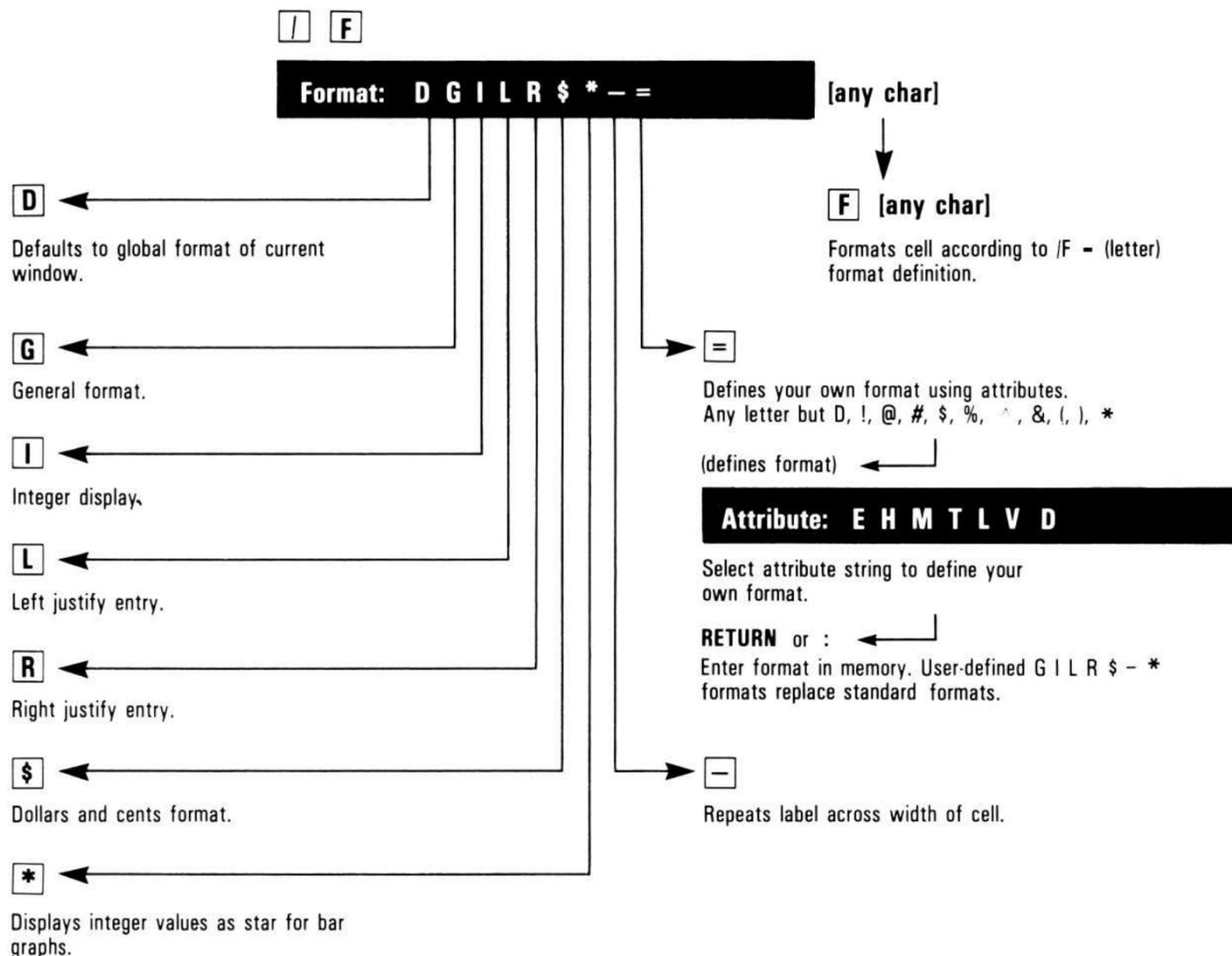
Note that you can scroll file names on the edit line with /SS, /SL, /SD, S#S, S#L, and /PF.

You can also use scrolling to modify a file name already in the directory—a handy technique for updating. For example, if you load and modify a file called SALES3.VC and want the update version to be called SALES4.VC, type /SS→.

Continue to press → until the name .D1/SALES3.VC appears on the edit line. Then erase the 3 by using **ESCAPE**, replace it with 4, and reappend the .VC suffix (or use CONTROL-E to edit the file name). Press **RETURN** to save the current worksheet as .D1/SALES4.VC.

If you want to modify a pathname, you can press ↑ to jump the edit cue back to the previous slash. For example, if you have the pathname .D1/SUB1/FILE1 and you want to change SUB1 to SUB2, you would press ↑ **ESCAPE ESCAPE**, then type 2/FILE1.

**FORMAT COMMAND—/F**



The Format command controls how an entry is displayed or printed. It applies to one location on the worksheet; the effect is called a *local format*. With the format command, a cell may have only one format assigned to it.

**Note:** The Format option of the Global command controls how the entire worksheet is displayed; the effect is called the *global format*. A local format overrides the global format. When the VisiCalc program is loaded (or cleared with the Clear command) the global format is General, and displays values to the maximum precision.

Each cell on the worksheet can be assigned a local format. The label or value in a cell is not changed by the Format command; only their appearance is affected. The entry line shows the value or label as it was entered. The full precision of a value is used in calculations.

The entry line shows the local format. The local format is assigned to the cell contents, not the cell location; if the contents are moved to another location, the local format moves with it. Erasing the contents of a cell with the Blank command, however, does not remove a local format. Clearing the worksheet with the /CA or /CS options of the Clear command removes all local formats.

To assign a format to a row or column before writing on it, format the first entry and replicate the format with the Replicate command across the row or down the column.

The following example is used to illustrate all the Format command options. To type this example and save it for future reference, follow these steps.

Example:

Type:	Result:
1. /CAY	The worksheet clears, the cursor moves to A1, and the global format is set to General.
2. Label entry → 1.23456789 → 99.999 RETURN	One label and two values are displayed.
3. /SSformex RETURN	The example is saved in a file named formex.

### **Graph Format — /F\***

The Graph format displays the number of asterisks equal to the truncated (not rounded) integer value of the cell. A cell can display one fewer asterisk than the column width (the first character space is blank).

Example:

<b>Type:</b>	<b>Result:</b>
1. /CAY	The worksheet clears and the cursor moves to A1.
2. 1↓2↓3↓4↓5 > B1RETURN	Puts a column of numbers in column A.
3. /F*	Sets the Graph format at B1.
4. /RRETURN↓.B5RETURN	
5. + ← RETURN	The formula + A1 sets B1 equal to A1.
6. /R RETURN↓.B5 RETURN R	Replicating the formula with a relative reference to A1 sets B2-B5 equal to A2-A5. Now the values are displayed as numbers in column A and a bar graph in column B.

Any number typed at A1-A5 is displayed as a number in column A and a bar of asterisks in column B. (A number larger than the column width displays the maximum number of asterisks, one less than the column width.) See "Drawing Bar Graphs" in Lesson Nine for more detailed examples of the graph function.

### **Repeating Labels — /F —**

You can repeat a label character across a cell by typing /F — and the character. This works the same way as the Repeating Label command (/—). See the section below on "The Repeating Label Command" for more information.

## Assigning Complex Formats Easily—/F=

The /F= option allows you to use a letter, or shifted number to stand for a complex format that you specify with the options of the Attribute command. You can specify up to 35 formats, any one of which you can invoke by typing /F and the format character.

You should follow these steps when defining your own format:

1. With the cursor in any cell, type /F=. The following prompt will appear:

```
User Format: Name: A-Z !`#$%&'()
```

2. Type a one-character format name. This name identifies the set of attributes you are about to enter as a format that can be invoked with /F.

3. After you enter the format name, this prompt will appear:

```
User Format: E H M T L V D O
```

This prompt lists the same options as the Attribute command prompt, plus O. You can now select the attributes you want for this format the same way you select attributes in the /A command (for information on attributes, see the preceding section in this chapter, "The Attribute Command").

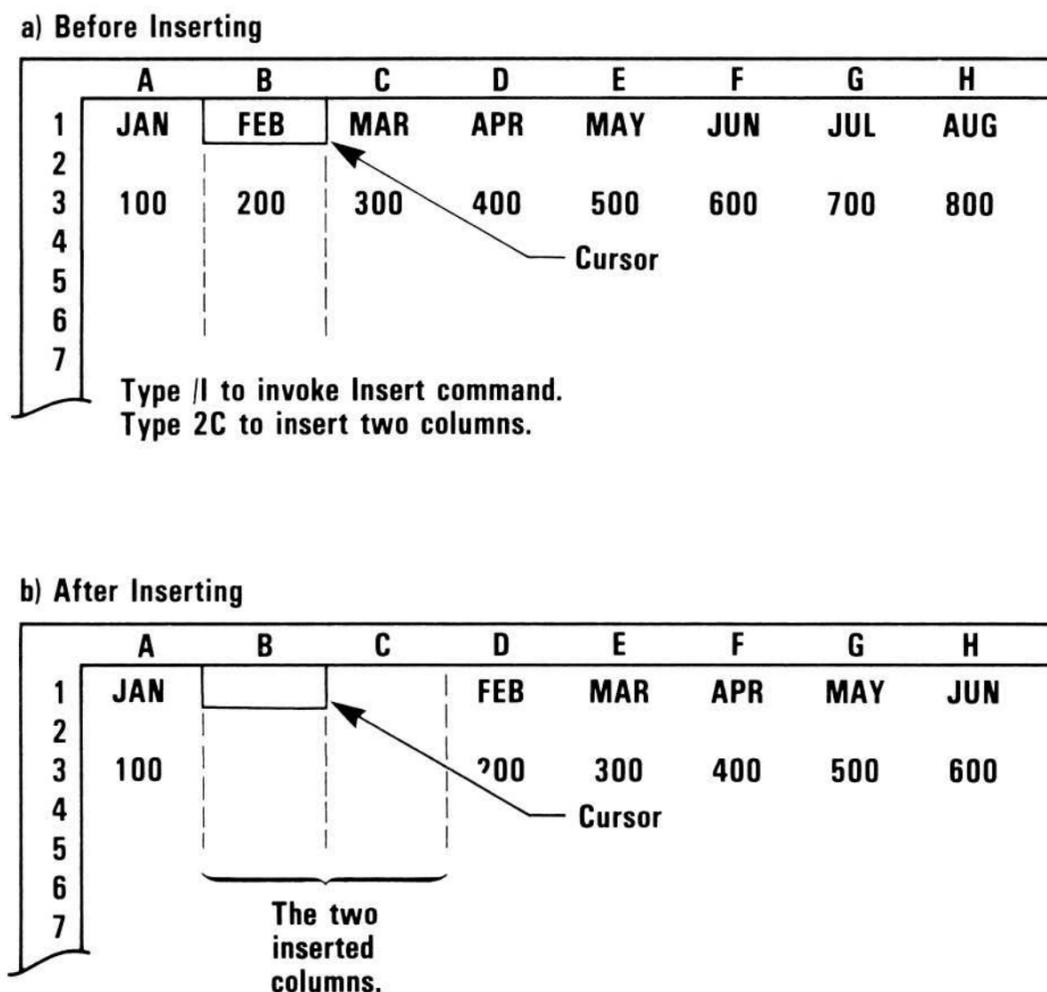
**Note:** A format does not automatically go into effect in the cell where the format was defined. To assign the format to the cell, type /F(format name).

Whenever you type /F followed by the name of a format you defined, the attributes identified with the format name are assigned to the cell where the cursor is.

You can name formats that you define with any letter (except D), or any shifted number (!`#\$%&'()). If you name a format I, L, R, \$, be aware that for the purposes of the worksheet where you defined it, your format takes the place of the program's standard format. If you name your format I, for instance, typing /FI will display the cell's contents according to your definition of I, not in integer format, as would usually be the case.

If you name a format I, L, R, \$, or \*, you can go back to the original program-supplied format by typing /F=(format name)OY. You can not name a format /FD, because this is the default format. If you want to clear a format that is named something other than I, L, R, \$, type /F=(format name)OY.

Figure 3-13 shows how columns are inserted. Inserting rows is similar.



010-022

Figure 3-13. Inserting Columns

If no integer precedes the R or C, one row or column is inserted.

One cell cannot be inserted, only an entire row or column.

When rows or columns are inserted into the range of an arithmetic function, the range is expanded to include the insertion. (Rows or columns cannot be inserted before the first location or after the last location of the range; the point of insertion must lie between the first and last locations.)

Because rows move down and columns move to the right to make space for an insertion, a row cannot be inserted if row 254 contains an entry; a column cannot be inserted if column BK contains an entry. To make an insertion, you must erase the contents of these cells (with the Blank command) or move them (with the Move command).

## Row Option — /IR

The Row option inserts blank rows. All rows from the first row in the insert range to the bottom of the worksheet move down to make room for the new rows.

All cell references in formulas on the worksheet are changed to correspond to the new coordinates of the rows that moved. For example, if a formula contains the coordinate C2 and one row is inserted above row 2, the coordinate in the formula is changed to C3.

Example:

<b>Type:</b>	<b>Result:</b>
1. /CAY	The worksheet clears and the cursor moves to A1.
2. Value → 1 → 2 → 3 RETURN > A2 RETURN Formula → +B1+C1 → +C1+D1 RETURN > A3 RETURN Sum → +B1+B2 → +C1+C2 RETURN	Entry Line: C3 (V) +C1+C2 The screen should look like figure 3-14.



010-030/P

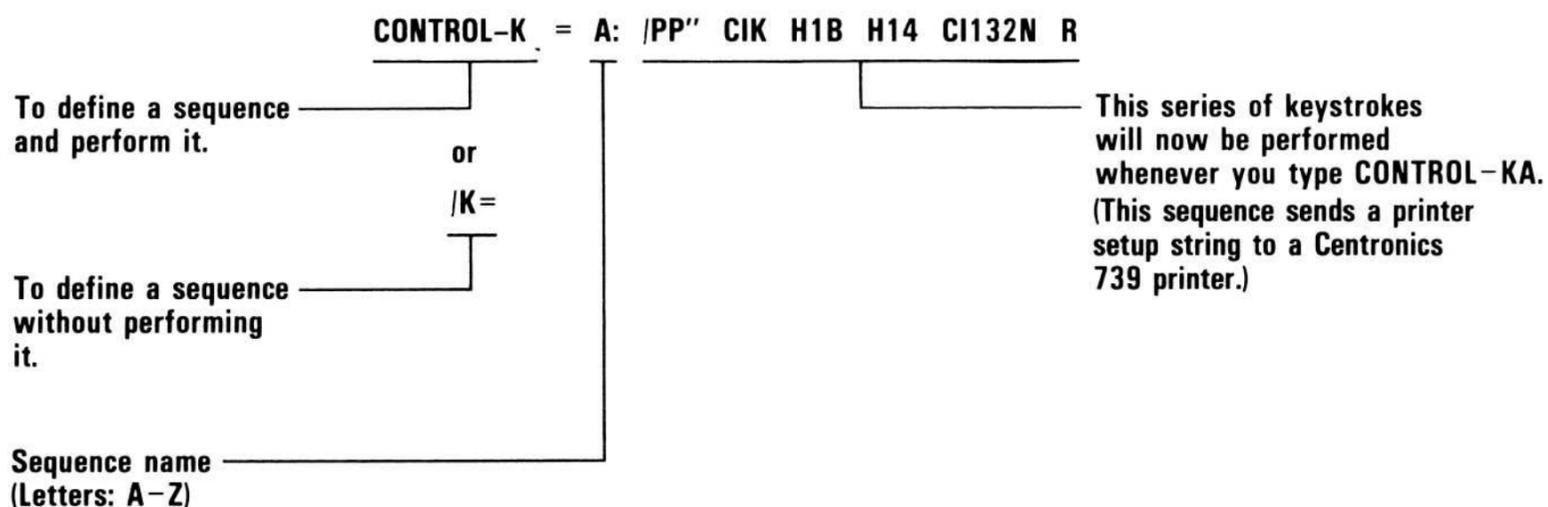
Figure 3-14. Worksheet Before Inserting Row

The command has four options:

- = Define a sequence
- C Delete all sequences
- E Edit a sequence
- RETURN Review all sequences

The following figure illustrates how the Keystroke Memory command works.

**Sequence:** A series of keystrokes that is assigned a one-character name. The keystrokes are performed automatically when you type CONTROL-K and the one-character name.



**Examples:**

Sequence Definition	Name	Strings
CONTROL-K =	A:	/PP" ^ CIK ^ H1B ^ H14 ^ CI132N ^ R
/K=	B:	JAN ^ > FEB ^ > MAR ^ > APR ^ > MAY ^ R
/K=	C:	/R: ^ > . ^ > ^ > ^ > ^ R

010-023

**Figure 3-16. Defining a Keystroke Sequence**

## The Define Sequence Option—/K= or CONTROL-K=

The Define Sequence option lets you determine the series of keystrokes you want to put into memory for using later and lets you assign the series a one-character name. You can define sequences that enter data, move to other cells, and use several commands to consolidate worksheets. The only valid sequence names are the letters A-Z.

There are two ways to define a sequence: /K=(sequence name) and CONTROL-K=(sequence name). CONTROL-K= allows you to define a sequence as you enter it on the worksheet; the keystrokes you type are executed as normal VisiCalc keystrokes *and* saved as a sequence. This allows you to try out different keystroke combinations before storing a sequence in memory. /K= captures the keystrokes you type without executing them.

A sequence can be defined at any time. After it has been defined and stored in memory, a sequence is invoked by typing CONTROL-K followed by its one-letter name. The only valid sequence names are the letters A-Z. Typing an integer after you type CONTROL-K but before you type a sequence name will execute the sequence that number of times. For example, CONTROL-K 5 B means you will execute the sequence named "B" 5 times.

The maximum size of a sequence is 125 characters. Because some keystrokes are represented within sequences as more than one character, you might not be able to fit 125 keystrokes in a single sequence. Using the chain-to-another-sequence character (^K) in a sequence allows you to execute multiple sequences. With this feature, you can overcome any limitation (except memory size) on the length of a sequence. You can split a long sequence into two shorter sequences, and chain the second to the first, to get the same effect. You can also type an integer after ^K and before the name of the sequence you are chaining. For example, typing "^K5A" at the end of a macro loads the sequence named A and executes it 5 times.

Setup strings set with /PSR are saved along with the worksheet where they were typed. If you want to use the same setup string on several worksheets, you can type the setup string while a blank worksheet is displayed. Then you load and save each worksheet you want the setup string to apply to, clearing the screen with /CSY before loading the next worksheet.

To change a setup string entered with the /PSR option, type /PSS. The setup string will appear on the edit line, with the edit cue highlighting the first character in the setup string. Use the normal VisiCalc edit keys (ESCAPE, →, ←, ↓, ↑, etc.) to edit the setup string. When you have finished editing the setup string, press RETURN. The new setup string will take effect; it will be saved with worksheet if you use the save command (/S).

Setup strings sent with the /PP'' option of the Print command are not saved in memory. If you type a setup string with /PSR and save it with the worksheet using /SS, then type another setup string when printing using /PP'', the setup string sent with /PP'' will be added to the setup string defined with /PSR. The /PSR setup string will still be in memory, however, and would still be saved with the worksheet.

## **PRINTING THE WORKSHEET IN SECTIONS**

If all /PS printer-setting options are set to N, and if the rectangle to be printed is wider than the number of characters the printer can print on one line, the part of the line beyond the printer limit is printed on a separate line. For example, if your printer prints 80 characters per line, the 81st character starts the next row down.

To reproduce a worksheet wider than the printer can print, print it in sections. Calculate how many columns of the worksheet can be printed at a time; then, divide the worksheet into the number of rectangles required to print the full width. The separate printed copies can be taped together to reproduce the worksheet.

If the printer uses single sheets rather than continuous-form paper, the worksheet might have to be printed in sections to accommodate the length.

You can also use the Printer Settings option of the Print command (/PS) to help you print your worksheet in sections. See the preceding section on "The Printer-Setting Options – /PS."

## Creating a Print File—/PF

The File option of the Print command (/PF) sends the image of the worksheet to a diskette file. A print file is a standard text file that contains exactly what is required to print the worksheet, including carriage returns and line feeds. It can be read by other programs, such as a word processor.

You might find it useful to add the suffix .PRF to the file name you specify. .PRF identifies the file as a print file, rather than a worksheet file, which should be given the suffix .VC or a DIF file which should be given the suffix .DIF. Because the file contains only the titles and numbers displayed on the worksheet (not the formulas, formats, etc., used to set up the worksheet), this file cannot be loaded with the Load option of the Storage command.

If you specify the name of a file that exists on the diskette, the VisiCalc program prompts you to make sure you want to replace the file: "File exists: Y to replace." If you type Y, the print file of the worksheet replaces the file with the same name on the diskette; if you type anything else, the Print command is canceled.

## DISKETTE FILES

Specifying a diskette file saves the image of the worksheet on diskette. See "File Names" in this chapter for a description of file names and how to specify them.

The following example shows how to save the area of the worksheet bounded by B10 at the upper-left and F20 at the lower-right in a print file named BUDGET.PRF on drive 1.

Example:

Type:	Result:
1. > B10	The cursor moves to B10.
2. /P	Prompt line: Print: File, Print, Settings The upper-left corner of the rectangle to be saved is defined as B10.
3. F	Prompt Line: Print: File name
4. budget.PRF RETURN	Prompt Line: Print: Lower right or "setup
5. F20 RETURN	Edit Line: F20 The worksheet is saved as a print file.

## Creating Subdirectories: The /SC Command

You might want to create a subdirectory to group together related files. A subdirectory is just like a file in the main directory, except that it contains one or more files or subsubdirectories.

A subdirectory is listed in the directory just like a file, except that its name ends with a slash. The files in the subdirectory are not listed in the directory—instead, they are listed in the subdirectory. Each subdirectory can have its own subsubdirectories, and their names are listed in the subdirectory just like files, except that they also end with slashes.

Insert your storage diskette in drive 2. If you have a one-drive system, follow the keystrokes below and switch diskettes when prompted to do so.

Example:

Type:	Result:
1. <b>/SC.D2/</b>	Prompt Line: Storage: Create subdirectory
2. <b>→</b>	Edit Line: .D2/
3. <b>SUBDIR1 RETURN</b>	The prompt and edit lines clear; your subdirectory named SUBDIR1 has been added to the directory.

To add a file to the subdirectory, first add information to a sheet so the file won't be blank (this makes it easier to verify that the procedure worked).

Example:

Type:	Result:
1. <b>/CAY SUBDIR1 RETURN</b>	Clears the worksheet, and puts SUBDIR1 at A1.
2. <b>/SS →</b>	Saves the file. Edit Line: .D2/SUBDIR1/ (If something else is on the edit line, press → until .D2/SUBDIR1/ appears.)
3. <b>FILE1.VC RETURN</b>	The drive whirs, and the prompt line clears.

To load the file just created:

Type:	Result:
1. /CAY	Clears the worksheet.
2. /SL →	Edit Line: .D2/SUBDIR1/ FILE1.VC
3. RETURN	The drive whirs, and the file loads. Position A1: SUBDIR1

To create a second subdirectory:

Type:	Result:
1. /CAY	Clears the worksheet.
2. SUBDIR2 RETURN	Position A1: SUBDIR2
3. /SC →	Edit Line: .D2/SUBDIR1/ When you scroll through the directory after entering /SC, only subdirectory names are displayed.
4. ESCAPE ESCAPE	Edit Line: .D2/SUBDI
5. 2 RETURN	Creates a subdirectory with the name SUBDIR2.

To edit names on the edit line another way, at step 4 above, press ↑. When you are scrolling through a directory, pressing ↑ backs up the edit cue to the previous slash (in this case, the slash after .D1). This option is particularly useful if you want a completely different subdirectory name. You can also type **CONTROL-E** to edit more extensively.

Now save the worksheet by typing:

**/SS → FILE2.VC RETURN**

To retrieve a file from a subdirectory:

Type:	Result:
1. /SL →	Scroll with → until the edit line reads .D2/SUBDIR1/
2. Press RETURN	Edit Line: .D1/SUBDIR1/ FILE1.VC

---

**VISICALC® Advanced Version****OPERATING THE APPLE III****WHAT YOU NEED**

The VisiCalc® Advanced Version program runs on the Apple III computer with at least 128K memory and the built-in disk drive. An additional drive is recommended. You can use either a floppy disk or a hard disk for storing your worksheets. You'll need a printer, unless you don't want to print your worksheets.

You also need the VisiCalc Advanced Version Loader and Program diskettes. If you want to use a floppy disk drive for storing your worksheets, you'll need at least one initialized diskette. The diskette **must** be initialized **before** you load the VisiCalc (AV) program. See the *Apple III Owner's Guide* for information on how to initialize a diskette. If you want to use a hard disk unit for worksheet storage, create at least one subdirectory for that purpose. See the section "Make a Subdirectory" in Chapter 5 of the *Apple III Owner's Guide* for directions.

**THE KEYBOARD**

Figure A-1 shows the keyboard. Most of the keys are self-explanatory, but a few need further description. Note the keys that are shaded; you will use them a great deal.

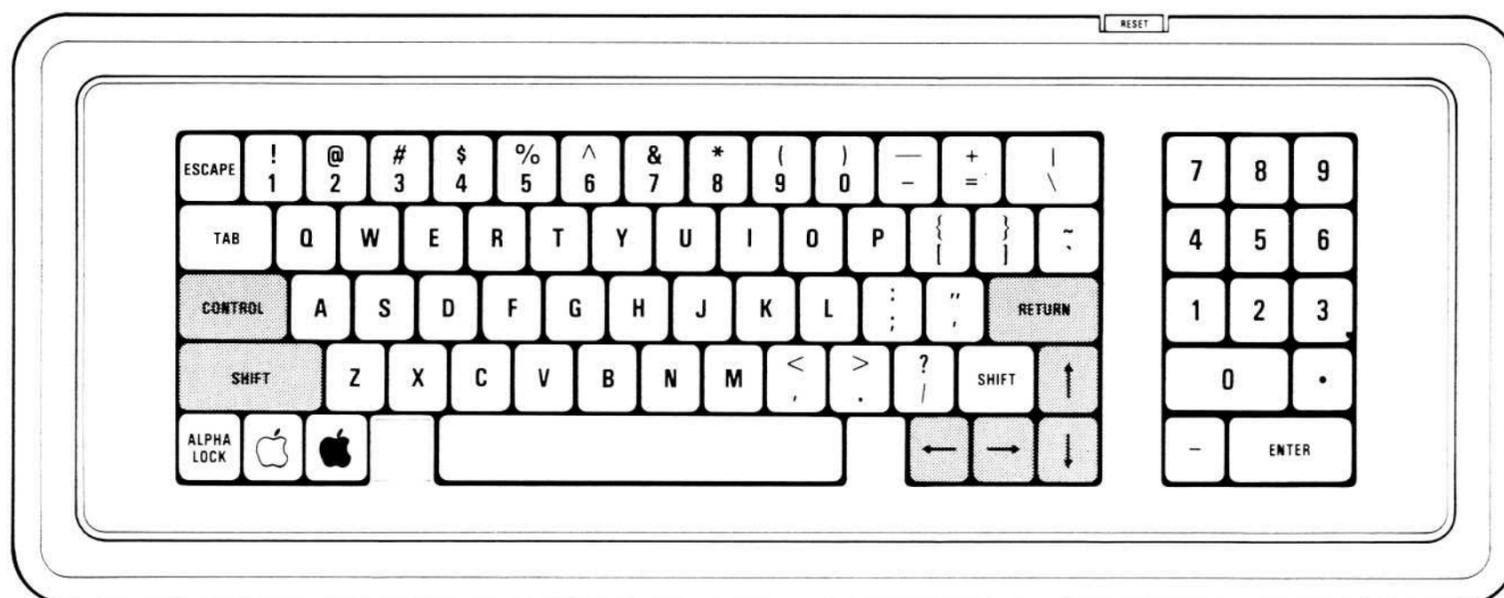
The keys marked with arrows move the cursor on the screen.

The RETURN key ends commands and worksheet entries.

The ESCAPE key backspaces to correct typing errors.

Holding down the CONTROL key and typing C cancels an entire entry or command; it also stops the printer or diskette drive when printing or saving a worksheet.

The Shift key works just like it does on a typewriter.



009-001

Figure A-1. Apple III Keyboard

## HANDLING DISKETTES

A diskette is a small, magnetically coated plastic disk, sealed in a protective square plastic cover. Through the oval cutout in the square cover, you can see the magnetic surface of the actual diskette.

*Never* touch the exposed magnetic surface with your fingers or any implement; you could lose data. Protect the diskette from dust by storing it in the paper sleeve it comes in. Keep it at least six inches from magnetic fields such as those generated by a TV. Extremes of temperature (such as in a car trunk on a warm day) can damage a diskette, destroying valuable data or your VisiCalc program. Don't bend, staple, or write on the square plastic cover with a hard pen or pencil (use only a soft felt-tip pen).

## LOADING THE VISICALC PROGRAM

First, make sure you have at least one freshly initialized diskette available. Follow the directions under "Formatting Blank Diskettes" in the *Apple III Owner's Guide*. If you are using a hard disk unit for storing your worksheets, create a subdirectory on the hard disk. See the section "Make a Subdirectory" in Chapter 5 of the *Apple III Owner's Guide* for directions.

As you face the front of the computer, the power switch is at the rear of the left side. With the power switch off, open the built-in diskette drive door. If you are using an external floppy disk drive for storing your worksheets, open that disk drive door, too. If a diskette is in either drive, remove it. With the drive door(s) open and the power off, follow this procedure:

1. Remove the VisiCalc Loader diskette from its paper sleeve and hold it by the label edge with the label side up. Carefully insert the diskette into the built-in drive with the label edge entering last (see Figure A-2). Gently push the diskette all the way in and close the drive door.

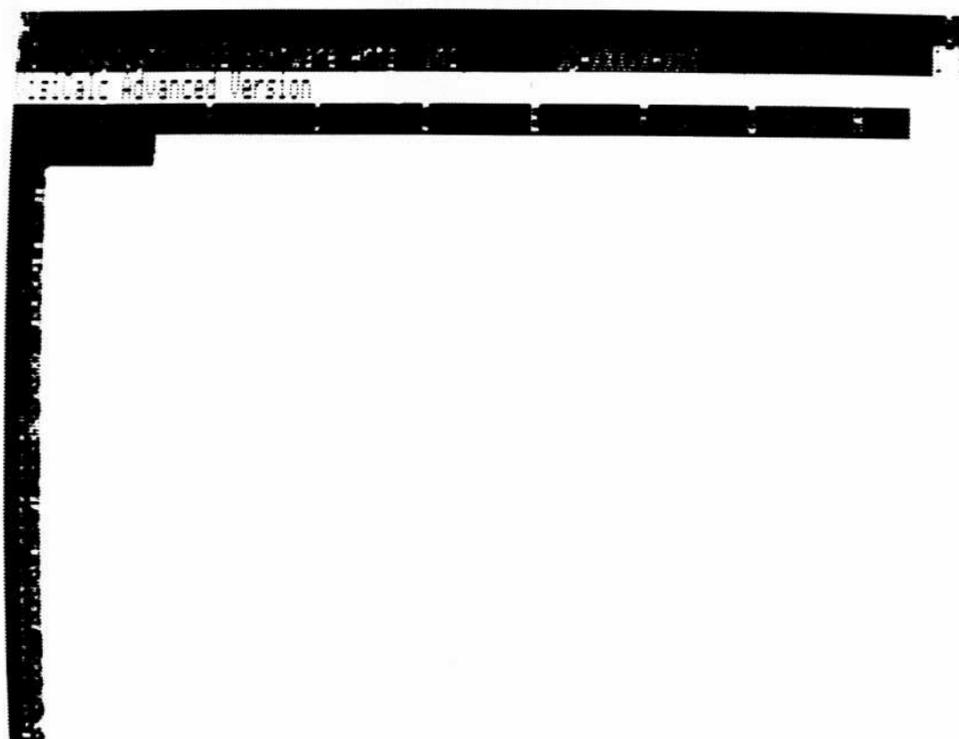


009-001/P

**Figure A-2. Inserting a Diskette**

2. Turn on the power switch. The drive motor whirs, and two copyright notices appear on your screen. Then the VisiCalc screen appears. The prompt line reads `:(C) Copyright 1982 Software Arts, Inc.`
3. Remove the VisiCalc Loader diskette from the built-in drive and put it back in its paper sleeve. Repeat step 1 with the VisiCalc Program diskette.

4. Press the **RETURN** key. After a few seconds, the VisiCalc copyright information appears on the screen. The screen should look like Figure A-3.



010-036/P

**Figure A-3. The VisiCalc Screen**

If the screen doesn't look like Figure A-3, remove the VisiCalc Program diskette from the drive. Turn off the computer and go through the load procedure again. If, after several tries, the VisiCalc program does not load, see your dealer for help.

5. After you have loaded the program successfully, keep the VisiCalc Program diskette in the built-in drive until you are finished using the program.
6. If you are using an external floppy disk drive, insert the initialized diskette into that drive and close the door. You are ready to use the VisiCalc program.

12. Pick up the Loader diskette. On the right edge of the diskette, near the top, is a thin piece of plastic with an adhesive backing, called a *write-protect tab*. (It looks like a piece of electrical tape.) Carefully peel the tab off and put it on your desk or someplace handy. Don't throw it away or stick it to anything; you will replace it on the diskette in just a minute. Try not to bend the diskette when peeling off the tape; you could damage it if you are not careful.
13. Type **/S#S .dl/vcswappath.dif**. Do not type RETURN yet.
14. Take the Program diskette out of the built-in drive (drive 1).
15. Put the Loader diskette (with the write-protect tab removed) into the built-in drive (drive 1) and close the drive door.
16. Press **RETURN**. The program prompts for the lower-right coordinate of the data area to be saved; type **A1 RETURN**. The next prompt reads *Data save: R C* or RETURN. Type **RETURN**. The DIF file will be saved on your Loader diskette.
17. Carefully replace the write-protect tab on the Loader diskette.
18. Put the Program diskette back in the built-in drive (drive 1).

Now all you have to do to load VisiCalc is turn on your ProFile™ and put the Loader diskette in drive 1. The program automatically looks for the overlay files on the ProFile.



**VISICALC® Advanced Version****FORMULAS USED TO CALCULATE THE FINANCIAL FUNCTIONS**

Financial functions perform financial calculations, such as Present Value, Future Value, and Internal Rate of Return.

The Financial functions are presented in the table below (arguments are separated by commas).

<b>Function</b>	<b>Result</b>
@RATE(n,pmt,pv,fv)	Interest or discount rate of the given arguments.
@PERIODS(i,pmt,pv,fv)	Number of periods of the given arguments.
@PMT(i,n,pv,fv)	Amount of each payment of the given arguments.
@PV(i,n,pmt,fv)	Present Value of an <i>even</i> series of payments.
@FV(i,n,pmt,pv)	Future Value of an <i>even</i> series of payments.
@IRR( <i>investment,range</i> )	Internal Rate of Return based on an initial <i>investment</i> and a <i>range</i> of even or uneven cash flows. Both arguments must be supplied.
@NPV(i, <i>range</i> )	Net Present Value of a series of even or uneven payments; this series is the <i>range</i> argument.

The formulas used in the financial functions are listed in Appendix C.

For the purposes of Financial functions, money paid out (payments or investments) has a negative value and all money received (loans or investment income) has a positive value. Negative interest rates indicate depreciation.

Values returned by the financial functions depend on the formulas used in the functions. In certain cases, a financial function must choose between two solutions of its formula. This might cause values on your worksheet to differ from values returned by a calculator or annuity table. In such a case, you might want to know the formula used to calculate a function. For this reason, the financial functions are listed below.

**NOTE:** Abbreviations for the arguments used by the Financial functions are listed below.

- n* Number of periods; must be an integer greater than zero.
- pmt* Payment each period; can be positive or negative.
- i* Interest rate; must be positive or negative but not zero.
- pv* Present value of a series of payments. May be positive or negative but not a range.
- fv* Future value of a series of payments. May be positive or negative but not a range.

### THE RATE FUNCTION-@RATE

When payment (*pmt*) is not equal to zero, @RATE is solved for *i* using an iterative method:

$$0 = pv * (i + 1)^u + pmt * (1 - (i + 1)^{-v})/i + fv * (i + 1)^{-v}$$

where *u* is the fractional part of *n* and *v* is the integer part of *n*.

If *pmt* is 0, the following formula is used:

$$i = (fv / -pv)^{(1 / n)} - 1$$

### THE PERIODS FUNCTION-@PERIODS

$$n = \ln((fv * i) - pmt) / (-p * (i - pmt)) \ln(i + 1)$$

### THE PAYMENT FUNCTION-@PMT

$$pmt = fv * i / (1 - (i + 1)^n) + pv * i * (i + 1)^n / (1 - (i + 1)^n)$$

### THE PRESENT VALUE FUNCTION-@PV

$$pv = (-pmt * (1 - (i + 1)^{-n})) / i + fv / (i + 1)^n$$

**APPENDIX D**

**FORMULAS USED TO CALCULATE THE FINANCIAL FUNCTIONS**

The Rate Function-@RATE .....	D-2
The Periods Function-@PERIODS .....	D-2
The Payment Function-@PMT .....	D-2
The Present Value Function-@PV .....	D-2
The Future Value Function-@FV .....	D-3
Net Present Value Function-@NPV .....	D-3
The Internal Rate of Return Function-@IRR .....	D-3

**GLOSSARY/INDEX**

Glossary/Index .....	Index-1
----------------------	---------

**FIGURE**

**TITLE**

1-1. The Electronic Worksheet .....	1-4
2-1. Empty Worksheet .....	2-3
2-2. Go To Command Prompt .....	2-7
2-3. An Automatic Recalculation .....	2-11
2-4. Calculating a Formula .....	2-13
2-5. Income as a Percentage of Sales .....	2-16
2-6. Branch Office Budget .....	2-20
2-7. Title and Column Headings of Budget .....	2-22
2-8. Making One Column Wider .....	2-23
2-9. Columns too Narrow for Values .....	2-25
2-10. Replicate Command Prompt .....	2-27
2-11. The Process of Replication .....	2-29
2-12. Replicating a Relative Reference .....	2-30
2-13. Single and Double Lines on the Worksheet .....	2-33
2-14. Expanded Budget .....	2-34
2-15. Invoice .....	2-41
2-16. Shipping Addresses of Invoice .....	2-44
2-17. Entry Line of Invoice .....	2-48
2-18. Screen Split Vertically .....	2-49
2-19. Invoice With Several Items .....	2-50
2-20. Totals Block of Invoice .....	2-51

2-21.	Invoice With Lookup Tables . . . . .	2-53
2-22.	Examples of Maximum Function . . . . .	2-56
2-23.	If Function Examples . . . . .	2-58
2-24.	Horizontal and Vertical Lookup Tables . . . . .	2-60
2-25.	Lookup Table for Retail Price . . . . .	2-61
2-26.	Lookup Table for Discount Percent . . . . .	2-63
2-27.	Printing the Formulas of the Worksheet . . . . .	2-73
2-28.	Invoice With Data Area To Be Saved . . . . .	2-76
2-29.	Data Area To Be Saved as a Separate File . . . . .	2-77
2-30.	Sales Report Worksheet for Data Files . . . . .	2-84
2-31.	Summary Sales Report With One Data Row . . . . .	2-86
2-32.	Income Variance Report . . . . .	2-93
2-33.	Inaccurate Variance Report . . . . .	2-96
2-34.	Financial Calculator . . . . .	2-100
2-35.	Future Value and Present Value Calculations . . . . .	2-103
2-36.	First Cost-Estimating Worksheet . . . . .	2-112
2-37.	First Cost-Estimating Worksheet . . . . .	2-113
2-38.	Second Cost-Estimating Worksheet . . . . .	2-116
2-39.	Graph of Monthly Sales . . . . .	2-118
3-1.	Comparing the Format and Attribute Commands . . . . .	3-6
3-2.	Attribute Settings . . . . .	3-7
3-3.	Local, Global and Startup Settings . . . . .	3-8
3-4.	Worksheet Displayed in Expression Format . . . . .	3-12
3-5.	Mode Option Settings . . . . .	3-14
3-6.	Gutters and Alignment in Cell Displays . . . . .	3-16
3-7.	Deleting Columns . . . . .	3-24
3-8.	A Reference Before Deleting a Column . . . . .	3-25
3-9.	A Reference After Deleting a Column . . . . .	3-26
3-10.	Lookup Tables . . . . .	3-49
3-11.	Recalculation by Column (Correct Result) . . . . .	3-67
3-12.	Recalculation by Row (Incorrect Result) . . . . .	3-68
3-13.	Inserting Columns . . . . .	3-73
3-14.	Worksheet Before Inserting Row . . . . .	3-74
3-15.	Worksheet After Inserting Row . . . . .	3-75
3-16.	Defining a Keystroke Sequence . . . . .	3-77
3-17.	A Pause in a Sequence . . . . .	3-84
3-18.	Moving Two Rows Downward . . . . .	3-93
3-19.	Moving Two Rows Upward . . . . .	3-94
3-20.	Moving Two Columns . . . . .	3-95
3-21.	Printer-Setting Options . . . . .	3-99
3-22.	Printer-Setting Pagination . . . . .	3-100

Financial functions take four arguments. Abbreviations for arguments to the Financial functions are presented in the following table.

n	Number of periods; must be a real number greater than zero.
pmt	Payment each period; can be positive or negative.
i	Interest rate; must be positive or negative but not zero.
pv	Present value of a series of payments. May be positive or negative but not a range.
fv	Future value of a series of payments. May be positive or negative but not a range.

The Financial functions are presented in the table below (arguments are separated by commas).

<b>Function</b>	<b>Result</b>
@RATE(n,pmt,pv,fv)	Interest or discount rate of the given arguments.
@PERIODS(i,pmt,pv,fv)	Number of periods of the given arguments.
@PMT(i,n,pv,fv)	Amount of each payment of the given arguments.
@PV(i,n,pmt,fv)	Present Value of an <i>even</i> series of payments.
@FV(i,n,pmt,pv)	Future Value of an <i>even</i> series of payments.
@IRR( <i>investment,range</i> )	Internal Rate of Return based on an initial <i>investment</i> and a <i>range</i> of even or uneven cash flows. Both arguments must be supplied.
@NPV(i, <i>range</i> )	Net Present Value of a series of even or uneven payments; this series is the <i>range</i> argument.

The formulas used in the financial functions are listed in Appendix D.

For the purposes of Financial functions, money paid out (payments or investments) has a negative value and all money received (loans or investment income) has a positive value. Negative interest rates indicate depreciation.

When a financial function takes periods ( $n$ ) and/or interest rate ( $i$ ) as arguments, they must be specified. This is true for @RATE, @PERIODS, @PMT, @PV, @FV, and @NPV.

Arguments must be values, formulas, or cell references. Of the arguments other than  $i$  and  $n$ , one can be @NA; all four may be entered if they are known. Two exceptions to this are the Internal Rate of Return Function and the Net Present Value function, which take only two arguments. Both arguments must be supplied.

The Financial functions calculate all values based on the *end* of the period. This is important for the @PMT, @PV, and @FV functions. To convert a result from payments made at the end of the period to payments made at the beginning of the period, multiply the result by  $(1 + i)$ .

### INTEREST OR DISCOUNT RATE—@RATE

The Interest or Discount Rate function calculates the interest rate (for loans) or discount rate (for annuities) based on the following arguments, in order: number of periods, payment each period, present value of the payments, future value of the payments.

The periods argument ( $n$ ) must be supplied. One of the following arguments may be entered as @NA if all other arguments are specified: payment (pmt), future value (fv), present value (pv).

Acceptable combinations of arguments are:

@RATE( $n$ ,pmt,pv,@NA)

@RATE( $n$ ,pmt,@NA,fv)

@RATE( $n$ ,@NA,pv,fv)

@RATE( $n$ ,pmt,pv,fv)

### **NUMBER OF PERIODS – @PERIOD**

The Number of Periods function calculates *n*, the number of periods required to satisfy the arguments given to the function. Its arguments are, in order: interest rate in decimal form, payment, present value, future value. This function returns a real number.

The interest rate argument (*i*) must be specified. One of the following arguments may be entered as **@NA** if all other arguments are specified: payment (*pmt*), future value (*fv*), present value (*pv*).

Acceptable combinations of arguments are:

@PERIODS(*i*,*pmt*,*pv*,**@NA**)  
@PERIODS(*i*,*pmt*,**@NA**,*fv*)  
@PERIODS(*i*,**@NA**,*pv*,*fv*)  
@PERIODS(*i*,*pmt*,*pv*,*fv*)

### **PAYMENT – @PMT**

The Payment function calculates the payment that must be made each period to satisfy the requirements of the arguments given to the function. These arguments are, in order: interest rate in decimal form, number of periods, present value, future value.

The periods (*n*) and interest rate (*i*) arguments must be specified. One of the following arguments may be entered as **@NA** if the other argument is specified: future value (*fv*), present value (*pv*).

Acceptable combinations of arguments are:

@PMT(*i*,*n*,*pv*,**@NA**)  
@PMT(*i*,*n*,**@NA**,*fv*)  
@PMT(*i*,*n*,*pv*,*fv*)

**FUTURE VALUE—@FV**

The Future Value function calculates the future value of a series of payments. Its arguments are, in order: interest rate in decimal form, number of periods, payment, present value.

The periods (n) and interest rate (i) arguments must be supplied. One of the following arguments may be entered as @NA if the other argument is specified: payment (pmt), present value (pv).

Acceptable combinations of arguments are:

@FV(i,n,pmt,@NA)

@FV(i,n,@NA,pv)

@FV(i,n,pmt,pv)

**PRESENT VALUE—@PV**

The Present Value function calculates the present value of a series of payments. The present value of a series of payments is a single amount of money at the present time that is the economic equivalent of the entire series.

The Present Value function takes as its arguments, in order: interest rate in decimal form, number of periods, payment, future value.

The periods (n) and interest rate (i) arguments must be supplied. One of the following arguments may be entered as @NA if the other argument is specified: payment (pmt), future value (fv).

Acceptable combinations of arguments are:

@PV(i,n,pmt,@NA)

@PV(i,n,@NA,fv)

@PV(i,n,pmt,fv)

**INTERNAL RATE OF RETURN — @IRR**

The Internal Rate of Return function returns an interest rate at which a discounted series of payments equals an initial investment. This function is a special case of the @NPV function. The @IRR function returns an interest rate for which the net present value of the discounted payments is equal to the negative of the initial investment.

This function takes two arguments, in order: investment (beginning value), and *range*. The investment argument can be either a cell or a value; it represents the initial investment for which you are calculating the rate of return. The *range* argument is a range of cells that contain the payments.

@IRR is the only financial function that can have multiple solutions. @IRR returns only the first rate at which NPV is 0, although there may be more than one. Appendix D, "Formulas Used to Calculate the Financial Functions," lists the formulas used to calculate the financial functions. If you get a different result from a financial calculator than you do with @IRR, the formulas being used may be different.

### **NET PRESENT VALUE – @NPV**

The @NPV function calculates the net present value of future cash flows. It takes two arguments, in order: the discount rate (or cost of money used to discount the future cash flows), and a range of locations that include the cash flows themselves.

## **Logic Functions**

A logical value is one whose value is either TRUE or FALSE. A logic function is one that manipulates logical values. Logical calculations are similar to mathematical operations, but operate only on this special set of values.

**Comparison Operators:** The comparison operators work on two numeric values and evaluate to a logical value. The equal (=) and not equal (< >) operators can be used to compare logical values. For example, the formula  $4 > 1$  (four is greater than one) evaluates to the logical value TRUE. The formula  $5 = 3$  (five is equal to three) evaluates to the logical value FALSE.

These comparisons can be used either as entries on the worksheet or arguments in a logic function. If used as entries on the worksheet, they should not be written at a location referenced by a function that does not use logic functions. If this happens, the value of such functions is ERROR.

The following table lists the comparison operators (it assumes the operator is preceded by *value1* and is followed by *value2*).

Operator	Value
<	TRUE if <i>value1</i> is less than <i>value2</i> ; FALSE if it is not.
>	TRUE if <i>value1</i> is greater than <i>value2</i> ; FALSE if it is not.
=	TRUE if <i>value1</i> is equal to <i>value2</i> ; FALSE if it is not.
< =	TRUE if <i>value1</i> is less than or equal to <i>value2</i> ; FALSE if it is not.
> =	TRUE if <i>value1</i> is greater than or equal to <i>value2</i> ; FALSE if it is not.
< >	TRUE if <i>value1</i> is not equal to <i>value2</i> ; FALSE if it is.

**@NOT:** @NOT takes a single logical value as its argument (i.e., one whose value is TRUE or FALSE); the function's value is the opposite logical value. The value of @NOT(A1) is FALSE if A1 is TRUE and TRUE if A1 is FALSE. If the value of A1 is @NA, the value of @NOT(A1) is also @NA. If the value of the argument is anything other than TRUE, FALSE, or @NA, the value of the @NOT function is ERROR.

**@AND:** @AND takes any number of arguments, each of which must be a logical value or a range of logical values. Its value is TRUE if all the arguments are TRUE, FALSE if any of the arguments is FALSE. If any of the arguments is not logical or evaluates to ERROR, the value of @AND is ERROR. If any of the arguments evaluates to NA and all other arguments are TRUE or FALSE, the value of @AND is NA.

**@OR:** @OR takes any number of arguments, each of which must be a logical value or a range of logical values. Its value is TRUE if any of the values is TRUE, and FALSE if all the values are FALSE. If any of the arguments is not logical or evaluates to ERROR, the value of @OR is ERROR. If any of the arguments evaluates to @NA and all other values are TRUE or FALSE, the value of @OR is NA.

**@IF:** @IF takes three arguments. The first must be a logical value; the second and third can be any value. The function evaluates to the value of the second or third argument, depending on the value of the first.

**THE FUTURE VALUE FUNCTION-@FV**

$$fv = pmt * (1 - (i + 1)^n) / i - pv * (i + 1)^n$$

**NET PRESENT VALUE FUNCTION-@NPV**

@NPV takes two arguments: interest or discount rate (*i*) and a range of cash flows (*cf1*, *cf2*, *cf3...cfk*).

$$npv = cf1 / (1 + i)^1 + cf2 / (1 + i)^2 + cf3 / (1 + i)^3 \\ \dots + cfn / (1 + i)^n$$

where *cf1* is the first cash flow in the range and *cfn* is the last cash flow in the range.

**THE INTERNAL RATE OF RETURN FUNCTION-@IRR**

@IRR takes two arguments: value at time 0 (*c*) and a range of cash flows (*cf1*, *cf2*, *cf3...cfn*).

The value at the beginning (*c*) is the initial investment and is typically a negative value.

This function is solved for *i* using an iterative method:

$$0 = cf1 / (1 + i)^1 + cf2 / (1 + i)^2 + cf3 / (1 + i)^3 \dots + cfn / (1 + i)^n + c$$

where *cf1* is the first cash flow in the range and *cfn* is the last cash flow in the range.

