micro
COMPUTER
books

John M. Nevison

EXECUTIVE COMPUTING

HOW TO GET IT DONE ON YOUR OWN

EXECUTIVE COMPUTING

HOW TO GET IT DONE ON YOUR OWN

JOHN M. NEVISON
John M. Nevison Associates
Concord, Massachusetts

Addison-Wesley Publishing Company
Reading, Massachusetts • Menlo Park, California
London • Amsterdam • Don Mills, Ontario • Sydney

This book is in the
Addison-Wesley Microcomputer Books
Executive Series

Seventh Printing, December 1983

The language BASIC was developed at Dartmouth College by John G. Kemeny and Thomas E. Kurtz.

VisiCalc™ is a trademark of Personal Software, Inc., and is a copyrighted computer program of Software Arts, Inc. Used with permission.

Excerpts from *Walden, The Writings of Henry D. Thoreau*, edited by J. Lyndon Shanley (copyright © 1971 by Princeton University Press), CEAA edn. pp. 10–323. Reprinted by permission of Princeton University Press.

Library of Congress Cataloging in Publication Data

Nevison, John M
  Executive Computing—how to get it done on your own.

  (Joy of computing)
  Includes bibliographical references and index.
  1. Business—Data processing—Case studies. 2. Computer programs—Problems, exercises, etc. I. Title.
II. Series.
HF5548.2.N415   658'.054   80-28977
ISBN 0-201-05248-2

ISBN 0-201-05248-2
GHIJKL-AL-8987654

Preface

To be a philosopher is not merely to have subtle thoughts . . . It is to solve some of the problems of life, not only theoretically, but practically.

Henry David Thoreau

This book is for the business person who wants to get work done with the help of a computer—from the busy executive who owns a microcomputer to the business student who needs a practical way to apply a quantitative method. In order to use this book comfortably, the reader should already have written and run a few BASIC programs (see Appendix A if BASIC is new to you).

TO HELP YOU HELP YOURSELF

The heart of this book is the idea that a business method can be expressed conveniently as a computer program. Each of the 25 cases tries to illustrate a fundamental business method and suggest some ways to apply it. The method is pulled together in the computer program at the end. Because a case must be the simple version of the story, its program is not really an end; it is a beginning—a place where you can start to journey toward the program you need to solve your own particular problem. So if you find a particular case especially interesting, start programming after you stop reading.

As you become familiar with the individual programs used to solve the various cases, you may elect to combine several to solve your own particular problem. Through modification and combination, you should be able to compose an approach to a wide variety of problems in a relatively short time.

For those who wish to test the waters before they jump in, exercises are included with each case. They are not difficult, but they do require more time than traditional exercises.

If you are interested in pursuing a subject beyond the depth of this book, the references at the end of each chapter should help you on your way.

The appendixes at the end of the book offer several kinds of additional help. The first provides a brief introduction to BASIC and a set of structural guidelines for writing and combining programs. The second includes a set of 16 utility programs to be used on their own or in larger programs. The third presents tips to help you buy a personal computer. The fourth describes another tool for manipulating tabular data.

BASIC

This book uses the computer language BASIC because it is:

- available on almost every computer
- quick to learn
- simple to read
- easy to understand
- powerful to use

The language BASIC is convenient, but not necessary. The ideas in the book are important; the computer language in which they are expressed is not. Whatever language you have available to you will work. BASIC is widely used and you should have no trouble finding a book that will explain the elements of the language (see Appendix A and its references). From there, the conversion to another language should be a straightforward task. If you attend to the principles of Appendix A, you should be able to transfer the ideas quickly and easily.

All of the programs in this book have been run on a Radio Shack TRS-80 computer with 16,000 characters of memory (16K) and Level II BASIC. This computer's successor, the Model III, presently sells for less than $1000. See Appendix C for additional details on equipment.

VISICALC

VisiCalc is a new computer product that turns a computer into an electronic blackboard on which you can write a table of numbers and modify it in many ways. It is an extremely handy way to deal with tabular information. Any program in this book that can be constructed using VisiCalc is marked with an asterisk (*) in the Program Index. See Appendix D for further details.

ACKNOWLEDGMENTS

It is a real pleasure to thank all those who have been involved in this book's evolution.

Teledata, Inc., of Hanover, New Hampshire, generously contributed computer resources that sped the writing and rewriting of the text and the programs. Business ideas and criticisms came from Michael Chu of City Investing, Lou Fernandez, Robert Montgomery of Corning, and Chris Nevison of Colgate University. Sharp comments on English came from Bruce Herzberg. Additional helpful remarks came from Bill Dickson, Tom Dwyer, and Steve Stadler. My colleagues at Nevison Associates, Tim Stein and Sandy Sorkin, improved both ideas and computer programs. Elisabeth Micheli painstakingly styled these programs for the TRS-80. Addison-Wesley's Bill Gruener and Gail Goodell provided unflagging support and encouragement. Barbara Pendergast meticulously edited the manuscript. The guest-cook problem was inspired by Toni McJennett. The largest debt of gratitude is again to Nancy Ross McJennett whose three performances as understanding wife, attentive mother, and concerned book designer made this book possible.

Quotes from Henry David Thoreau's *Walden* introduce each chapter with a remark that is intended to relieve the text's narrow focus.

J.M.N.

Concord, Massachusetts
December 1980

Contents

PART I THE EXECUTIVE OFFICE

APPENDIXES

PART I

THE EXECUTIVE OFFICE

1

Introduction: Looking at Numbers

The first ice is especially interesting and perfect, being hard, dark, and transparent, and affords the best opportunity that ever offers for examining the bottom where it is shallow.

Henry David Thoreau

Steven Cauldwell knew there were several advantages to writing his own computer programs, but he had no idea which one was the most important. He knew that writing a program forced him to express ideas in a different way and that just the effort of reexpression often taught him something new. For example, he had learned how to rework the simple idea of compound growth into a tool for dealing with pricing in inflationary periods.

COMPOUND GROWTH

Compound growth was a simple idea by itself. Growth at 15 percent was the same as multiplying by a factor of 1.15. In other words, the new sum would be 1.15 times the old sum. In BASIC that was:

```
LET S = 1.15 * S
```

The line read backwards. The initial S was on the right-hand side of the equal sign. The new S on the left-hand side was the result. When he made a four-line program and numbered each statement, the computer would do the statements in order:

```
10      LET S = 100
20      LET S = 1.15 * S
30      PRINT S
40      END
```

and the answer would appear as

```
 115
```

The initial idea had a simple consequence. To compound interest for five years, the same operation could be repeated five times. By using the NEXT statement to send the computer back to the FOR statement, he made the computer loop back and reuse the same statements:

```
10      LET S = 100
20 '
30      FOR Y = 1 TO 5
40          LET S = 1.15 * S
50          PRINT Y, S
60      NEXT Y
70 '
80      END
```

(Lines 20 and 70 are blank lines with an apostrophe mark. They do nothing but make the program easier to read.) The computer used the old statements with new values each time it went around the loop. The results were:

```
 1        115.
 2        132.25
 3        152.088
 4        174.901
 5        201.136
```

PRICING IN INFLATION

Pricing in an inflationary period was an elaboration of the idea behind compound growth. But different costs each had different inflation rates. Labor costs went up at a 1.15 rate each year, while raw materials went up at a 1.03 rate. If the dollar price contained 56 cents of raw material and 21 cents of labor, what happened to the profit margin when the price went up at a rate of 1.12 a year?

The program to answer the question was:

```
270        LET P = 100
280        LET R = 56
290        LET L = 21
300        PRINT  0; R, L, R+L, P, (P-(R+L))/P * 100
310        FOR Y = 1 TO 5
315            LET P = 1.12 * P
320            LET R = 1.03 * R
330            LET L = 1.15 * L
340            PRINT  Y; R, L, R+L, P, (P-(R+L))/P * 100
350        NEXT Y
999        END
```

The result was:

```
YR        MAT     LABOR  TOTL CST     PRICE  MARGIN

0        56.00    21.00    77.00     100.00    23%
1        57.68    24.15    81.83     112.00    27%
2        59.41    27.77    87.18     125.44    30%
3        61.19    31.94    93.13     140.49    34%
4        63.03    36.73    99.76     157.35    37%
5        64.92    42.24   107.16     176.23    39%
```

The price growth rate of 1.12 was clearly high, so Cauldwell retyped the price growth using 1.10 and reran the program. The margin was still too high. He quickly tried 1.08, 1.06, and finally 1.07 to find the value that preserved his margin.

In going from the compound growth to the pricing model, Cauldwell learned that an idea expressed as a computer program was almost always capable of refinement. He could, if he chose, go still further and make a model with five or six cost components. So the first big advantage of computing was the freedom to reexpress and refine an idea.

The computer program illustrated a second benefit: ease of calculation. It was easier and faster for Caudwell to write and revise the computer program than to find the answer any other way. With algebra he always worried about making an error; with his calculator he would not have seen the whole picture.

There was a third benefit: ease of reuse. The most obvious reuse was when the rates changed—and they were always changing. Now that the program was written, it could be saved and reused quickly when circumstances demanded a revised plan.

A fourth benefit was related to the ease of reuse: the ability to ask "what if?" questions. A computer program could be viewed as a set of assumptions, and what the computer did was draw the necessary conclusions in a fraction of a second. So if he wanted to use INFLATE to find out what a different future inflation rate would mean to his margin, he could. In fact, by trying out a few cases, he could develop his intuition about what the future might hold, both its threat and its promise. That was certainly a big advantage of computing.

THE PROGRAM INFLATE

```
100 REM   INFLATE   1 JANUARY 1980   STEVE CAULDWELL
110 '
112 REM   © COPYRIGHT 1980 JOHN M. NEVISON ASSOCIATES
114 '
120 REM   REVEALS PROFIT MARGIN IN AN INFLATIONARY
130 REM   WORLD WHERE RAW MATERIAL COSTS, LABOR COSTS,
140 REM   AND PRICES EACH GROW AT A DIFFERENT RATE.
145 '
150 REM   VARIABLES:
154 REM     L............LABOR COST
155 REM     P............PRICE
156 REM     R............RAW MATERIAL COST
158 '
200       PRINT “YR   RAW MAT.”, “LABOR”, “TOTAL COST”,
210       PRINT “PRICE”, “MARGIN (%)”
211       PRINT
220 '
270       LET P = 100
280       LET R = 56
290       LET L = 21
300       PRINT  0; R, L, R+L, P, (P-(R+L))/P * 100
310       FOR Y = 1 TO 5
315           LET P = 1.12 * P
320           LET R = 1.03 * R
330           LET L = 1.15 * L
340           PRINT  Y; R, L, R+L, P, (P-(R+L))/P * 100
350       NEXT Y
999       END
```

If you have never seen a BASIC computer program before, you should pause here and read Appendix A. If you have read programs before, then this one should hold no surprises for you.

The program uses three variables, R for raw materials, L for labor, and P for price. First the program prints out the headings for the results. The calculations then begin, and after printing the beginning values at time zero, the program prints out each year's values for five successive years.

To try a different price increase, you only have to retype the line LET P = 1.12 * P with a new rate.

Exercises

1. Verify that a price increase of 1.07 holds the margin constant. Is it exactly the right answer?
2. Suppose raw materials costs were 40 cents, energy costs were 16 cents, and labor costs were 21 cents. If the costs grow as before and energy costs grow at a rate of 1.35, what happens to the margin if the price increases at 1.15? Revise INFLATE to include the energy costs.

GRAPHING SALES

Steven Cauldwell was the Chief Executive Officer of Chordata, and his days were filled to overflowing. Just before he left the office one night, he learned that a meeting that was supposed to be held at the end of the week had been moved up to eight thirty the next morning. The meeting was with Frank Bradshaw, the director of the Bear Division. Bear Division made household appliances. It was the company's biggest division and its sales had not been doing well. It was nine o'clock at night. Cauldwell's problem was that he had only the rest of the evening to go over the figures and find out what had been happening.

The figures he had in front of him were the monthly sales for the last five years. He also had similar sales figures for each of Chordata's other three divisions. Cauldwell recalled a graph he had seen in a recent article that he felt would give him a quick yet comprehensive means of looking at the sales figures. The graph had two parts: a bar-chart summary of the average sales for several past years, and a line graph of the last twelve months' sales. It looked like the one shown in Fig. 1.1.

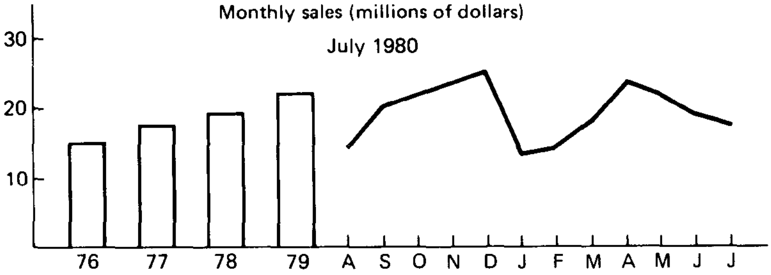

Fig. 1-1 A Sales Summary