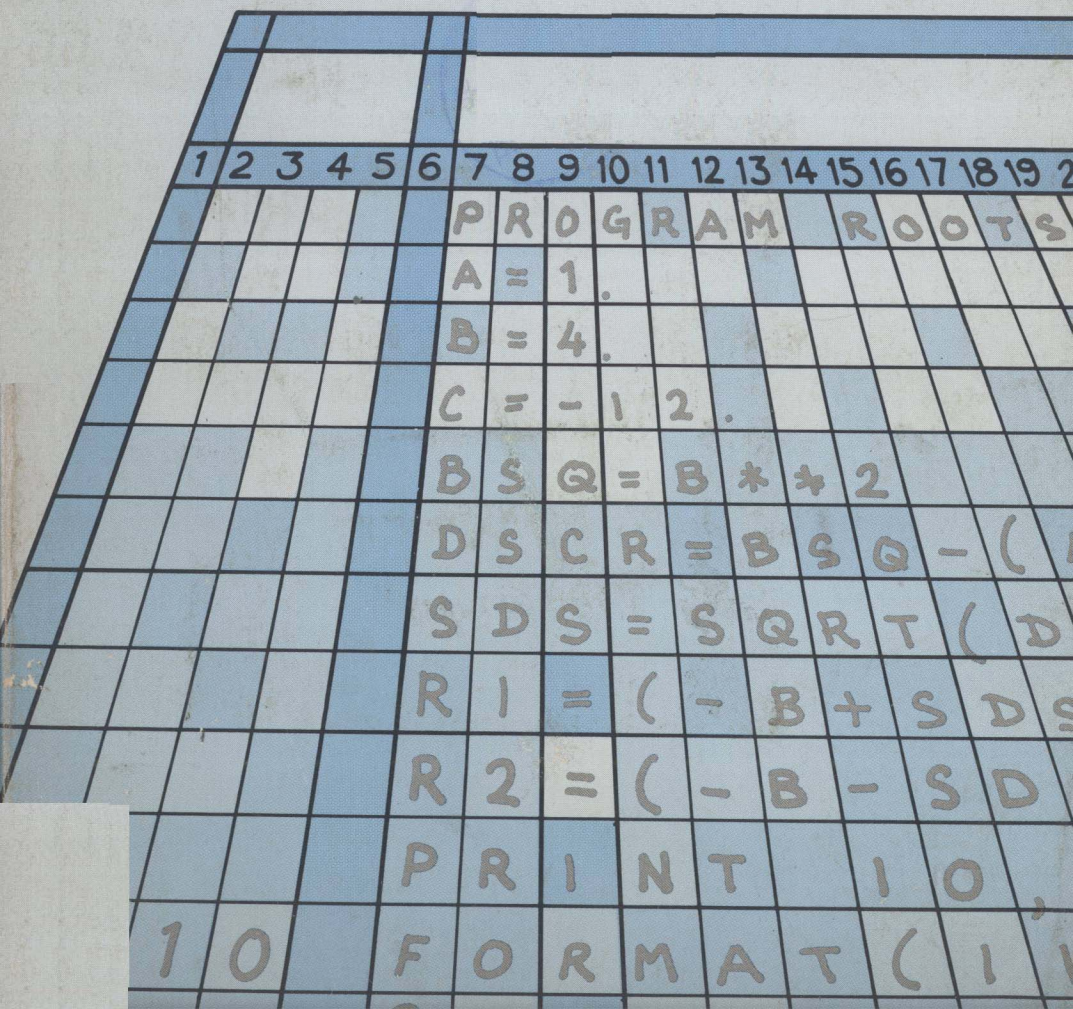


BEGINNING FORTRAN

Joe W. McKinley



1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
						P	R	O	G	R	A	M		R	O	O	T	S	
						A	=	1	.										
						B	=	4	.										
						C	=	-	1	2	.								
						B	S	Q	=	B	*	*	2						
						D	S	C	R	=	B	S	Q	-	(
						S	D	S	=	S	Q	R	T	(D				
						R	1	=	(-	B	+	S	D	S				
						R	2	=	(-	B	-	S	D					
						P	R	I	N	T									
1	0					F	O	R	M	A	T	(

Beginning FORTRAN

Joe W. McKinley

Professor

Department of Aerospace Engineering
California State Polytechnic University, Pomona

Matrix Publishers, Inc.
Beaverton, Oregon • U S A



DEDICATION

This book is dedicated to the memory of Barbara Ann Edwards.

©Copyright, Matrix Publishers, Inc., 1980

All rights reserved. No part of this book may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage and retrieval system, without written permission from the publisher.

10 9 8 7 6 5 4 3

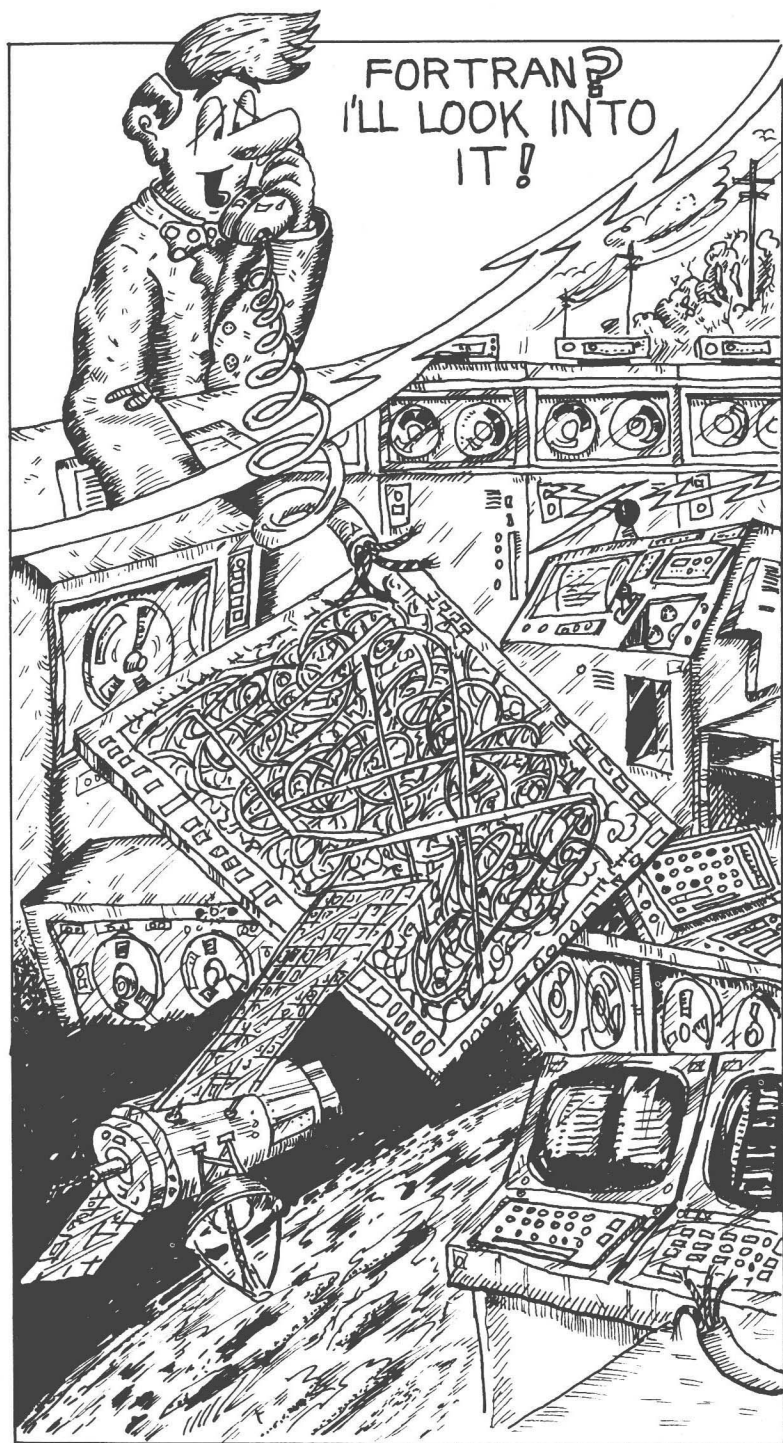
ISBN: 0-916460-11-8

Library of Congress catalog card number: 80-81871

Production: Patricia Miller
Illustrations: Scientific Illustrators
Printing: Pantagraph Printing
Editor: Merl K. Miller

Matrix Publishers, Inc.
11000 S.W. 11th Street
Beaverton, Oregon 97005

Beginning FORTRAN



Preface

This book has been generated out of the author's experience of teaching FORTRAN to students at the beginning level in college. It is intended for students in high schools, community colleges, colleges and universities who are beginning to learn the language of FORTRAN.

This book is primarily written about a batch processing mode of operation; however, people who are learning FORTRAN in an interactive or time-share mode will find the book extremely useful in conjunction with the local rules for operating their system.

No book can cover everything or be completely self contained. The book is aimed at the novice. Its goal is to give him, in a streamlined, relevant fashion, the material which is necessary for him to begin programming, and to help him to assimilate the skills needed to grow on his own. Upon the completion of the first two sections of this book, this goal should have been accomplished and the base for advanced study should be established. At this time the student can pursue the projects in the back of the book and the suggested references so as to proceed in his own way to the mastery of FORTRAN which he or she desires.

A key feature of the book is a streamlined, no nonsense approach whereby the student participates in the beginning and intermediate phases of programming directly to the use of the advanced considerations of the computer example projects. There is much emphasis on learning by doing through the execution of simple programs. There are also end of chapter problems and it is recommended that all of these be done by the student. The material of sections I and II form a base

whereby the student may go on into specialization in advanced level programming in any chosen field. Simple problems and projects form the basis for learning in Chapters 1 through 6 and it is in these chapters that the base is established. Then examples are given from a wide variety of engineering, scientific, mathematical, business and social science areas, as well as general numerical analysis and computer utilization, so that one may move into the study of his or her particular area of interest. Material in related field such as key punching, flow charting, and matrix algebra is given in the appendices.

The author would like to express his appreciation to the many students who have inspired this book. The help of Doris Jane McKinley in the course of the author's career is also appreciated as is that of his children, Frank and Celia. In particular, the invaluable assistance of Glenda S. Menges as production assistant and typist is appreciated. Many of the programs were written and run by Mr. Dan Newell. Several other Cal Poly students who contributed to the programming effort are: Richard Miranda, Jim Eshleman, Michael McMaster, and Julie Gardner. The editorial assistance of Prof. Paul Chirlian is gratefully acknowledged. The author would like to express further appreciation to California State Polytechnic University, Pomona, and especially to the Computer Center.

Joe W. McKinley
Pomona, California
January, 1980

Contents

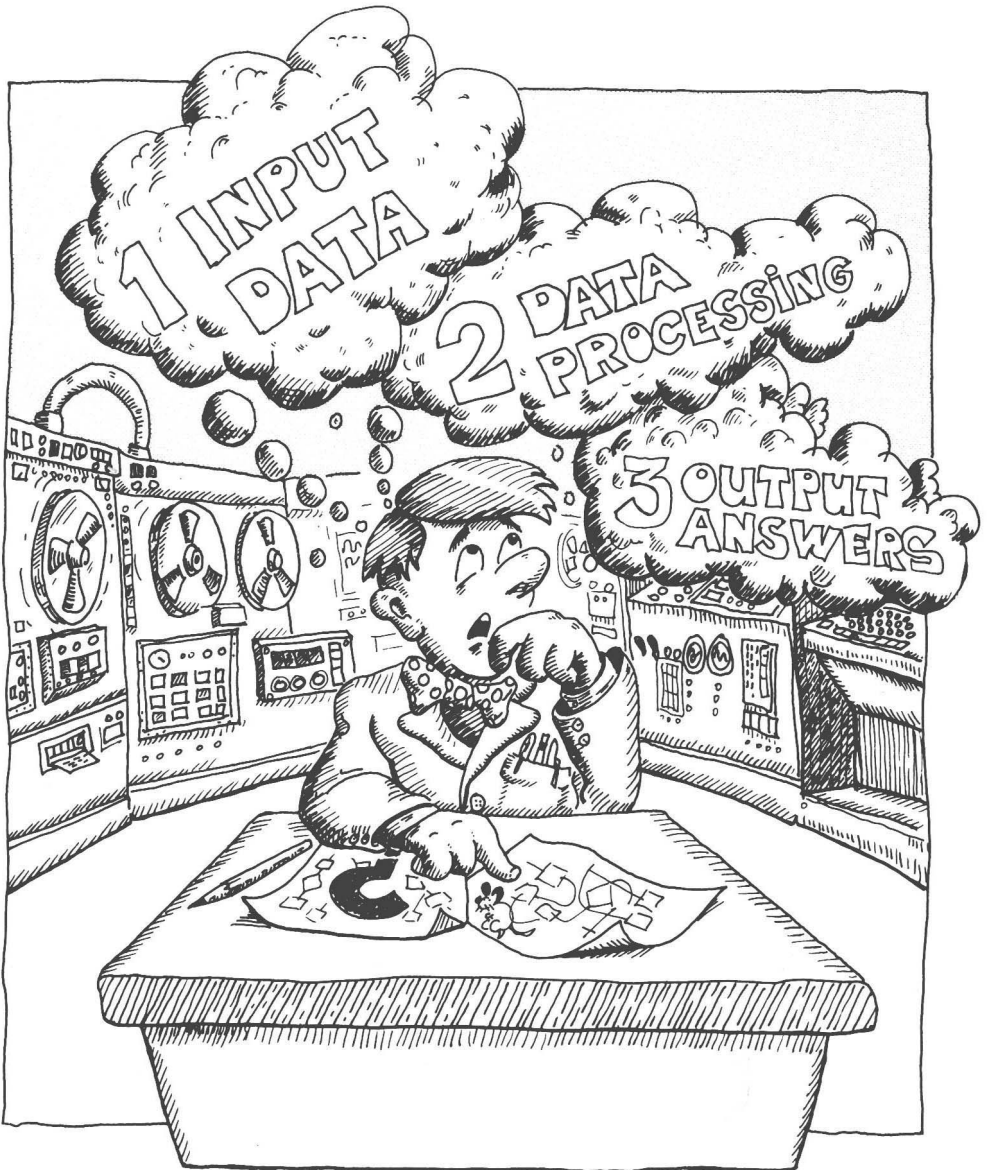
Section I Beginning Programming	1
1 Introduction: An Overview	3
Your First Program—Punch It and Run It Now	17
Features of Programming	17
A Final Overview Statement	22
2 Arithmetic Operations in FORTRAN	23
Integer and Real Constants and Variables	23
Arithmetic Operations, Expressions and Replacement	24
Arithmetic Assignments (Arithmetic Replacement)	26
Mixed Mode	27
Conclusion	29
3 Input/Output in FORTRAN	32
Input	32
Output in FORTRAN	39
Section II Intermediate Programming	47
4 Control Statements in FORTRAN	49
GO TO and Conditional GO TO	49
Assigned GO TO	50
Arithmetic IF	51
DO Loops	52
Logical IF	59
STOP	60
Sample Programs	61

5 Subscripted Variables	65
Introduction	65
Arrays	67
Subscript Forms	69
DIMENSION and COMMON	69
Implied DO Input/Output	74
6 Subprograms: Functions and Subroutines	85
Introduction	85
FUNCTION Subprograms	86
Multiple Values Back from the "FUNCTION" and a Light Introduction to "COMMON"	91
Statement Functions	94
Subroutines	95
"COMMON" Arrays and Variables in Subroutines	99
Multiple Returns from Subroutines	101
Other Allowable Calls/Subroutine Arguments	101
FORTRAN Library Functions and Computer Center Subroutines	102
Section III Advanced Programming	113
7 Some of the Fancier Tricks	115
Introduction	115
"TYPE" Declarations	115
"DATA" Declarations	116
Advanced "COMMON" and "EQUIVALENCE"	116
External	118
A New Trick for Headings	118
Multiple "ENTRY"	119
A-FORMAT and Character I/O	120
H-FORMAT and Variable Titles	121
Other Variable Types	121
Advanced Replacement—Mixed Mode	121
Section IV Computer Example Projects	125
Introduction	125
Project 1. Roots of a Polynomial	129
Project 2. Tabulation of Numerical Calculations	133
Project 3. Matrix Operations	137
Project 4. Grading of Students and Sorting into Ascending Order	141
Project 5. Solution of Linear Algebraic Equations (Simultaneous Equations)	151

Project 6.	Using the Line Printer as a Plotter	155
Project 7.	Study of Cost of Ownership of an Automobile	159
Project 8.	Mean, Standard Deviation and Correlation Coefficient	163
Project 9.	S-N Curve for Metal Fatigue	165
Project 10.	Lift from Wind Tunnel Data	169
Project 11.	Relativistic Velocity and Mass	175
Project 12.	Histogram and Histogram Plotting	179
Project 13.	Polynomial Curve Fit Program	183
Project 14.	Deflection of Beams	187
Project 15.	Enthalpy of Vaporization (Chemistry)	191
Project 16.	Calculation of Lattice Parameters from X-Ray Data	193
References	195
Appendix A	Keypunching	197
Appendix B	Flowcharting	207
Appendix C	Deck Set Ups	213
Appendix D	Introduction to Matrix Algebra	215
Appendix E	Scientific Subroutine Package	229
Answers to Selected Problems	235
Index	251

Section I

Beginning Programming



Chapter 1

Introduction: An Overview

This text is written to teach people to utilize the power of a computer in a specific way, namely writing a FORTRAN program to solve a problem in a batch processing environment. The text will also be extremely useful to those people who are using an interactive or time share mode, if it is used in conjunction with the execution instructions of the local facility. FORTRAN is a particular computer language of wide usage. There are, to be sure, other languages—BASIC, COBOL and ALGOL, to name a few—but FORTRAN ranks very high in popularity.

A batch processing environment means that the user will punch cards and leave them at the computer center. After a period of time, his punched cards will be returned to him along with a listing of his program and a listing of his answers. The time which elapses between the programmer's leaving his deck and the answers' being available for pickup is called the *turn-around time* of the computer center operation.

Let us imagine a programmer who wants to solve a simple example problem by using a computer. The programmer wants to solve for the roots of

$$ax^2 + bx + c = 0 \qquad (1.1)$$

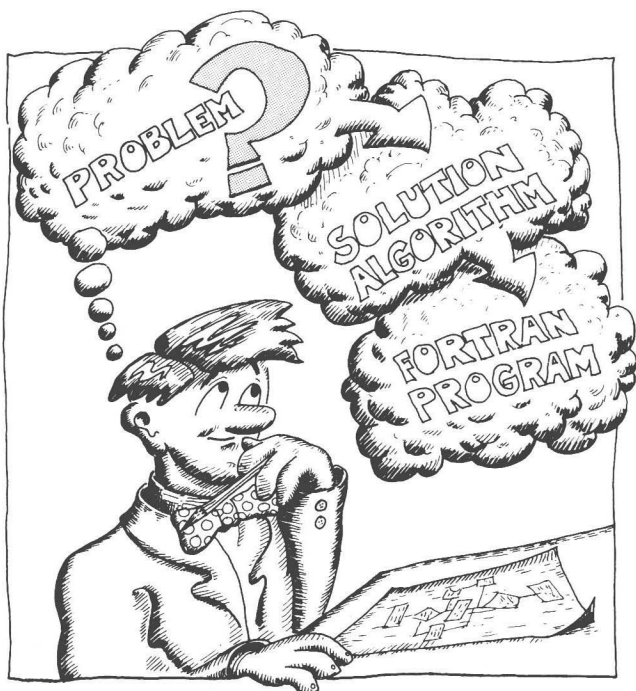
for some particular values of a , b , and c , for example,

$$\begin{aligned} a &= 1 \\ b &= 2 \\ c &= -12 \end{aligned}$$

Our programmer knows that the roots of this polynomial are given by

$$x_{1,2} = -b \pm \sqrt{b^2 - 4ac} / 2a \quad (1.2)$$

subject to some restrictions, which we shall ignore for the present time. Equation (1.2) is referred to as an *algorithm*. An algorithm is a computational procedure for getting a numerical answer. The programmer is now in the position of being able to write a program to solve for the answers he wants as depicted in Fig. 1.1.



A handwritten listing of such a program is shown in Fig. 1.2. Do not be concerned about all of the unknown details about this program. They will be explained soon. For now let us say that this is a short, simplified program, which can be improved as we continue to learn to talk to the computer in FORTRAN.

The programmer now punches this program on cards. The punched cards are shown in Fig. 1.3*. Again, we need to defer our complete understanding of these cards until later. A few other cards, called control cards, are now added to the FORTRAN deck. A schematic of

*See Appendix for instructions on keypunching.

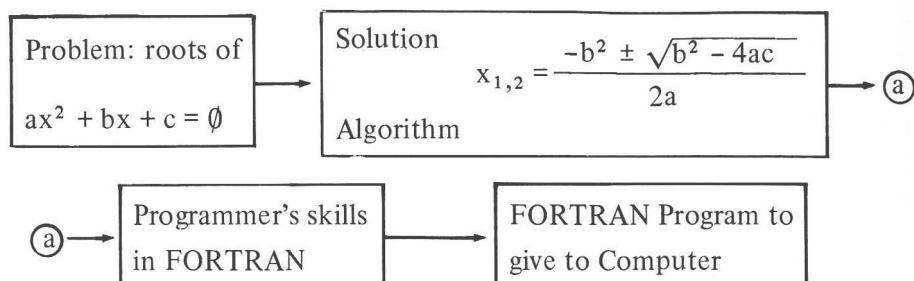


Figure 1.1 The programmer who knows the problem, and *how* to solve the problem, writes a FORTRAN program so that the computer will do the work.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31		
						P	R	O	G	R	A	M	R	O	O	T	S															
						A	=	1.																								
						B	=	4.																								
						C	=	-1	2.																							
						B	S	Q	=	B	*	2																				
						D	S	C	R	=	B	S	Q	-	(4.	*	A	*	C)											
						S	D	S	=	S	Q	R	T	(D	S	C	R)													
						R	1	=	(-	B	+	S	D	S)	/	(2.	*	A)										
						R	2	=	(-	B	-	S	D	S)	/	(2.	*	A)										
						P	R	I	N	T		1	0	,	R	1	,	R	2													
						1	0		F	O	R	M	A	T	(1	H	1	,	1	,	5	X	,	2	F	5	.	1)		
						S	T	O	P																							
						E	N	D																								

Figure 1.2 The handwritten listing of the programmer's FORTRAN program

his deck setup is shown in Fig. 1.4 and a sample deck is shown in Fig. 1.5. *Note that control cards are different for different computer centers!* The programmer then takes his deck to the computer center, or to some remote site, where it is either picked up or input directly and, in due course, he receives his results. The results for the example program are shown in Fig. 1.6.

