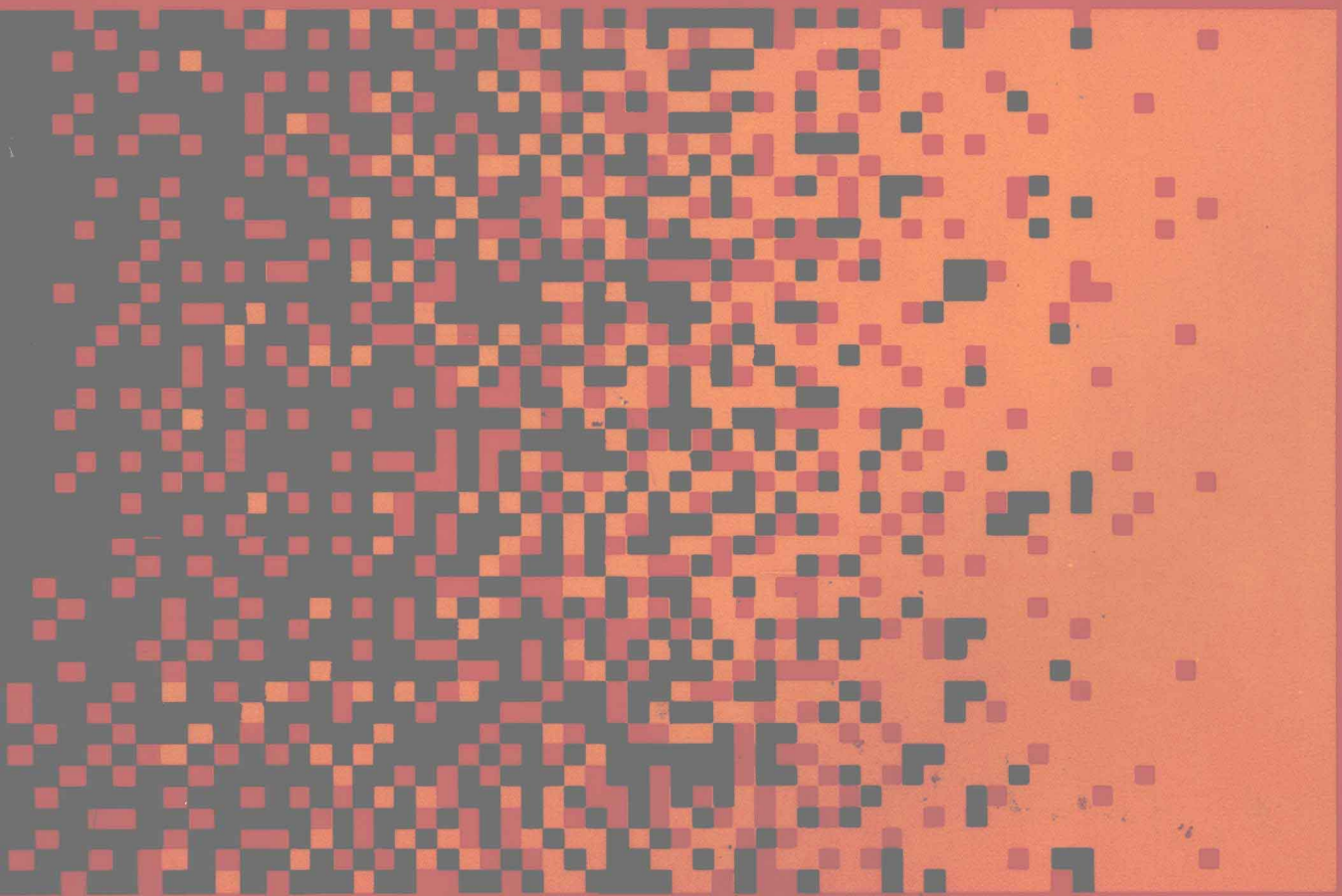
Third Edition

Fundamentals of FORTRAN 77 Programming

A Structured Approach

ROBERT C. NICKERSON



Fundamentals of FORTRAN 77 Programming

A Structured Approach

Third Edition

Robert C. Nickerson

San Francisco State University

HarperCollins*Publishers*

Library of Congress Cataloging in Publication Data

Nickerson, Robert C.
Fundamentals of FORTRAN 77 programming.

(Little, Brown computer systems series)
Rev. ed. of: Fundamentals of FORTRAN programming. 2nd ed. c1980.
Includes index.
1. FORTRAN (Computer program language) I. Nickerson, Robert C., 1946– . Fundamentals of FORTRAN programming. II. Title. III. Title: Fundamentals of FORTRAN seventy-seven programming. IV. Series.
QA76.73.F25N49 1985 001.64'24 84-25111

ISBN 0-316-60653-7

Library of Congress Catalog Card No. 84-25111

ISBN 0-673-39039-X

9 8

KPF

Printed in the United States of America

Previous editions of this book were entitled *Fundamentals of FORTRAN Programming.*

Disclaimer of Liabilities: Due care has been exercised in the preparation of this book to ensure its effectiveness. The author and publisher make no warranty, expressed or implied, with respect to the programs or other contents of this book. In no event will the authors or publisher be liable for direct, indirect, incidental, or consequential damages in connection with or arising from the furnishing, performance, or use of this book.

Fundamentals of FORTRAN 77 Programming

A Structured Approach

Little, Brown Computer Systems Series

CHATTERGY, RAHUL, AND UDO W. POOCH
Top-down, Modular Programming in FORTRAN with WATFIV

COATS, R. B., AND A. PARKIN
Computer Models in the Social Sciences

CONWAY, RICHARD, AND DAVID GRIES
An Introduction to Programming: A Structured Approach Using PL/I and PL/C, third edition
Primer on Structured Programming: Using PL/I, PL/C, and PL/CT

CONWAY, RICHARD, DAVID GRIES, AND E. CARL ZIMMERMAN
A Primer on Pascal, second edition

CRIPPS, MARTIN
An Introduction to Computer Hardware

EASLEY, GRADY M.
Primer for Small Systems Management

FINKENAUR, ROBERT G.
COBOL for Students: A Programming Primer

FREEDMAN, DANIEL P., AND GERALD M. WEINBERG
Handbook of Walkthroughs, Inspections, and Technical Reviews: Evaluating Programs, Projects, and Products, third edition

GRAYBEAL, WAYNE, AND UDO W. POOCH
Simulation: Principles and Methods

GREENFIELD, S. E.
The Architecture of Microcomputers
The Architecture of Microcomputers, Volume I: Fundamentals

GREENWOOD, FRANK
Profitable Small Business Computing

HEALY, MARTIN, AND DAVID HEBDITCH
The Microcomputer in On-Line Systems: Small Computers in Terminal-Based Systems and Distributed Processing Networks

LEMONE, KAREN A., AND MARTIN E. KALISKI
Assembly Language Programming for the VAX-11

LIAS, EDWARD J.
Future Mind: The Microcomputer—New Medium, New Mental Environment

LINES, M. VARDELL, AND BOEING COMPUTER SERVICES COMPANY
Minicomputer Systems

MARCA, DAVID
Applying Software Engineering Principles

MASHAW, B. J.
Programming Byte by Byte: Structured FORTRAN 77

MILLS, HARLAN D.
Software Productivity

MONRO, DONALD M.
Basic BASIC: An Introduction to Programming

MOSTELLER, WILLIAM S.
Systems Programmer's Problem Solver

NICKERSON, ROBERT C.
Fundamentals of Programming in BASIC
COBOL Programming
Fundamentals of FORTRAN Programming
Fundamentals of FORTRAN Programming: A Structured Approach, third edition

PARIKH, GIRISH
Techniques of Program and System Maintenance

PARKIN, ANDREW
Data Processing Management
Systems Analysis

PIZER, STEPHEN M., WITH VICTOR L. WALLACE
To Compute Numerically: Concepts and Strategies

REINGOLD, EDWARD M., AND WILFRED J. HANSEN
Data Structures

SAVITCH, WALTER J.
Abstract Machines and Grammars

SCHNEIDERMAN, BEN
Software Psychology: Human Factors in Computer and Information Systems

SHIVA, SAJJAN G.
Computer Design and Architecture

WALKER, HENRY M.
Problems for Computer Solutions Using BASIC
Problems for Computer Solutions Using FORTRAN

WEINBERG, GERALD M.
Rethinking Systems Analysis and Design
Understanding the Professional Programmer

WEINBERG, GERALD M., AND DENNIS GELLER
Computer Information Systems: An Introduction to Data Processing

The Little, Brown Microcomputer Bookshelf

BANSE, TIMOTHY

Home Applications and Games for the Apple® II, Apple® II Plus, and Apple® IIe Computers
Home Applications and Games for the ATARI® 400™/800™, 600XL™, 800XL™, 1200XL™, 1400XL™, and 1450XLD™ Home Computers
Home Applications and Games for the Commodore 64®
Home Applications and Games for the IBM® PC and PCjr
Home Applications and Games for the VIC-20

BARNETT, MICHAEL P., AND GRAHAM K. BARNETT

Personal Graphics for Profit and Pleasure on the Apple® II Plus Computer
Personal Graphics for Profit and Pleasure on the IBM® Personal Computers

HODGES, WILLIAM S., AND NEAL A. NOVAK

Personal Finance Programs for Home Computers

LEMONE, KAREN A.

Assembly Language and Systems Programming for the IBM® Personal Computer

MORRILL, HARRIET

BASIC for IBM® Personal Computers
Mini and Micro BASIC: Introducing Applesoft® , Microsoft® , and BASIC PLUS

NAHIGIAN, J. VICTOR, AND WILLIAM S. HODGES

Computer Games for Businesses, Schools, and Homes
Computer Games for Business, School, and Home for TRS-80 Level II BASIC

ORWIG, GARY W., AND WILLIAM S. HODGES

The Computer Tutor: Apple® Computer Edition Learning Activities for Homes and Schools
The Computer Tutor: ATARI® Home Computer Edition (for the ATARI® 400/800™, 600XL™, 800XL™, 1200XL™, 1400XL™, and 1450XLD™ Home Computers)
The Computer Tutor: IBM® Personal Computer Edition (for the IBM® PC and PCjr)
The Computer Tutor: Learning Activities for Homes and Schools (for the TRS-80® , Apple® , Commodore 64™, VIC-20, and PET / CBM® Home Computers)

SIMPSON, TOM, AND SHAFFER & SHAFFER APPLIED RESEARCH & DEVELOPMENT, INC.

VisiCalc® Programming: No Experience Necessary

SHAFFER & SHAFFER APPLIED RESEARCH & DEVELOPMENT, INC.

VisiCalc® Programming: No Experience Necessary, for the Apple® II, Apple® II Plus, and Apple® IIe Personal Computers
VisiCalc® Programming: No Experience Necessary, for ATARI® 800™ and 1200XL™ Home Computers
VisiCalc® Programming: No Experience Necessary, for the IBM® Personal Computer
VisiCalc® Programming: No Experience Necessary, for the TRS-80® Model III Microcomputer

SHNEIDERMAN, BEN

Let's Learn BASIC: A Kids' Introduction to BASIC Programming on the Apple® II Series
Let's Learn BASIC: A Kids' Introduction to BASIC Programming on ATARI® Home Computers
Let's Learn BASIC: A Kids' Introduction to BASIC Programming on the Commodore 64®
Let's Learn BASIC: A Kids' Introduction to BASIC Programming on IBM® Personal Computers

WINDEKNECHT, THOMAS G.

6502 Systems Programming

Preface

The objective of this book is to provide a carefully paced introduction to structured computer programming and the FORTRAN 77 language for students with a minimal mathematical background. To accomplish this objective the book systematically introduces the features of FORTRAN 77 as they are needed for various processing situations. These language features are illustrated by program examples that are drawn from nonmathematical problems that most readers will readily recognize. Structured programming concepts are developed along with, not separate from, the language features. As a result, the reader not only learns the FORTRAN 77 language but also gains an understanding of the need for and use of each language element, and learns how to develop well-structured programs in FORTRAN 77.

The book adheres strictly to FORTRAN 77 (the 1978 version of ANS FORTRAN). List-directed input and output are introduced at the beginning. Block IF statements are used for decision structures. Loop control is accomplished by standard features in FORTRAN 77. Character data processing and file processing are explained in detail. Appendix A summarizes the FORTRAN 77 language elements covered in the book. The only exception to the strict FORTRAN 77 orientation is the last section of Chapter 5 which describes WHILE loops as implemented in WATFIV-S and some other versions of FORTRAN. In addition, Appendix B briefly discusses several other versions of FORTRAN and provides a table that compares features in different versions of the language.

Language features are covered in conjunction with structured program development. Basic program structure is discussed early with complete chapters on decision logic (Chapter 4) and loop control (Chapter 5). Nested decisions and loops are described fully in these chapters. The program development process is covered in detail in Chapter 7. Topics such as program refinement, design, testing, and documentation are discussed at appropriate points. Program style and understandability are emphasized throughout the book. The use of subroutines in the top-down design and development of large programs is covered in Chapter 11. Upon completion of this book the reader should be able to develop FORTRAN 77 programs that are well structured, understandable, and correct.

The first chapter of the book introduces the basic concepts necessary to understand programming and FORTRAN 77. Chapters 2 through 6 cover the fundamental elements of the language and develop basic programming

methodology. These chapters discuss input and output, arithmetic processing, decision logic, and loop control. Chapter 7 can be thought of as a capstone for these first chapters because it brings together many concepts of program development and explains them in detail. Chapter 7 is also a transition to the more advanced topics covered in Chapters 8 through 12. These chapters describe character and logical data processing, arrays, subprograms, and files (sequential and direct). The material in these advanced chapters can be read in different sequences. In addition, some of the advanced topics can be covered along with earlier chapters. (The chapter prerequisite structure is described completely in the instructor's manual.)

A number of features make the book especially useful. These include the following:

- The first section of Chapter 1 discusses essential computer concepts. This serves as an introduction to these topics for readers with no previous computer background or as a review for readers with some computer experience.
- The book emphasizes program development using interactive computer systems. Consequently, the book can be used with most micro, mini, and mainframe computers. Punched card program development is described briefly for those using this approach.
- The book is designed so that programming can begin as early as possible. After finishing Chapter 2, the reader can write complete programs of his or her own design. After each succeeding chapter, the reader can develop increasingly more complex programs.
- Many examples and illustrative programs are provided throughout the book. The examples are nonmathematical in nature and oriented toward applications the reader should easily understand. Complete lists of input data and output are shown with most sample programs.*
- Interactive program design is discussed in detail. Characteristics that make an interactive program easy to use are explained and illustrated.
- Many common algorithms are described, including algorithms for sequential and binary searching, sorting, sequential file updating, and direct file processing.
- Each chapter contains questions to review the material covered in that chapter. The answers to approximately half of the review questions are found in Appendix G.
- All chapters except the first contain a substantial number of programming problems. Most problems require only a minimal mathematical background and emphasize nontechnical areas, including business and social science. Some problems are designed for math,

* All names of persons, companies, and organizations in examples, problems, and questions in this book are fictitious and are used for illustrative purposes only.

science, and engineering students. The programming problems range in difficulty from relatively easy to very difficult and challenging. Test data is provided with most problems.

- Flowcharts are discussed in Appendix D. This allows this topic to be covered at the most appropriate time. The discussion in the appendix parallels the text development of the corresponding programming topics. All flowchart examples are keyed to illustrative programs in the book.
- Numerous appendices are included. In addition to those already mentioned, there are appendices covering FORTRAN-supplied functions (Appendix C), internal data representation (Appendix E), and exponential form and double precision data (Appendix F). The appendices can be used in various ways depending on the reader's needs.

An instructor's manual is available that contains teaching suggestions, course schedules, chapter objectives, lists of terms, chapter outlines, answers to review questions, and test questions and answers. Also included in the instructor's manual are overhead transparency masters for a number of illustrations from the book.

Many of the ideas for this edition of the book came from reviews by users of the second edition. I greatly appreciate their effort. The manuscript reviewers did an excellent job and their comments were especially useful. I would like to thank Robert E. Case, United States Military Academy; Steve Drasner, North Virginia Community College; Henry A. Etlinger, Rochester Institute of Technology; Maurice D. Lind, Jefferson State Junior College; Ronald D. Schwartz, Baldwin Wallace College; and Judith Watson, Virginia Polytechnic Institute, for their participation in the reviewing process. Many of their suggestions have been incorporated into the book.

Finally, I would like to thank my family for their support and help during the writing of this book.

Contents

Chapter 1

Introduction to FORTRAN programming

A computer is a device that is used to solve problems. The process that a person goes through in instructing a computer how to solve a problem is called programming. Programming involves combining words and symbols that are part of a special language. FORTRAN is a language that is commonly used for programming solutions to many types of problems.

This book is about programming in the FORTRAN language. The book describes the main rules of FORTRAN and explains the general process of computer programming. It also presents many programming examples for different types of problems. As a result, you should not only learn the fundamentals of the FORTRAN language, but also you should gain an understanding of the programming process and an insight into different computer applications.

Chapter 1 introduces the basic concepts necessary to begin studying FORTRAN. The first section covers elementary computer concepts. We then introduce the FORTRAN language and describe the general process of programming in FORTRAN. After completing this chapter you should have the background needed to begin learning to program in FORTRAN. Later chapters go into detail about the FORTRAN language, the programming process, and computer applications.

1-1. Computer concepts

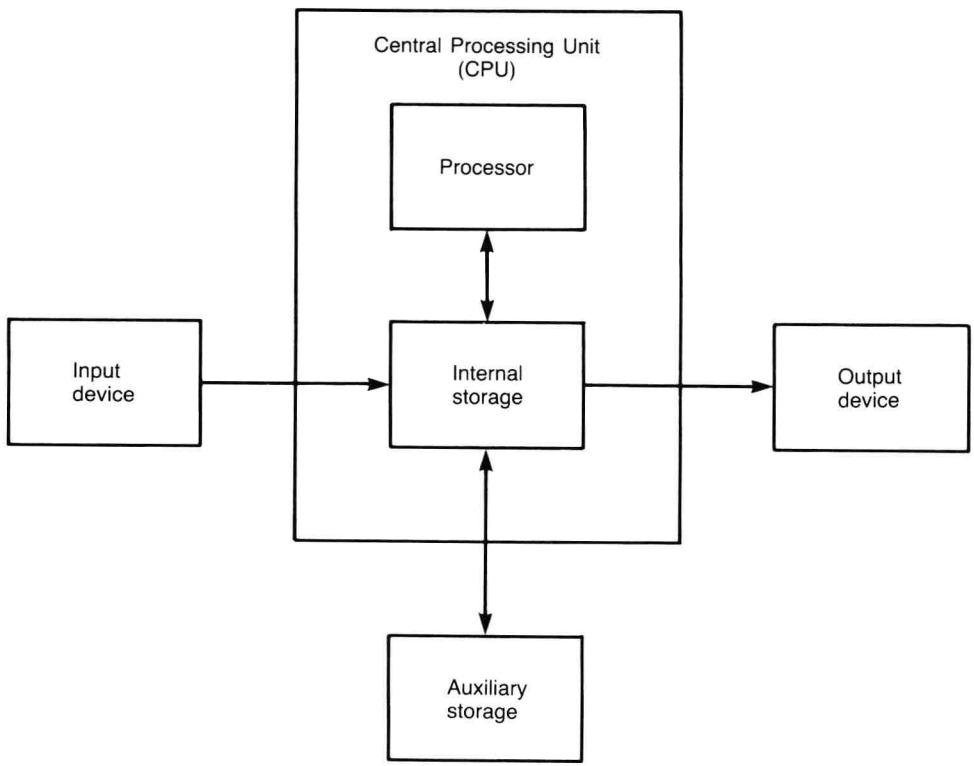
Three topics should be understood before studying FORTRAN: computers, programs, and data. Basically, a *computer* is an electronic device that processes data by following the instructions in a program. A *program* is a set of instructions that is stored in the computer and performed automatically by the computer.

*Data** is facts, figures, numbers, and words that are stored in the computer and processed according to the program's instructions.

Computers

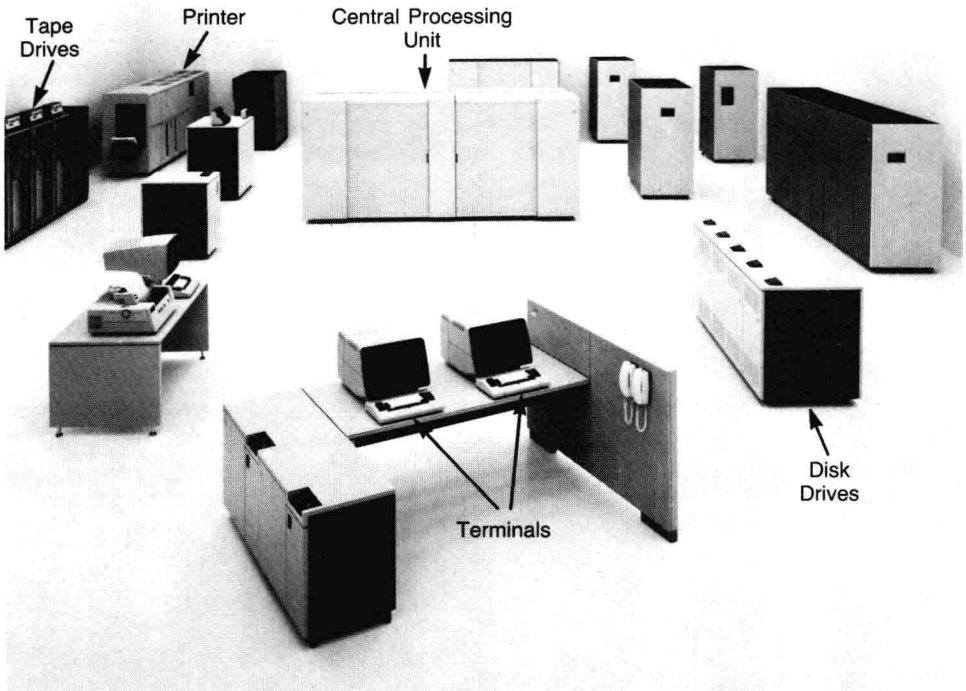
A computer consists of several interconnected devices or components. One way to view the organization of a computer is shown in Figure 1-1. In this diagram, boxes represent the different components of the computer and lines with arrowheads show the paths taken within the computer by data and program instructions. There are five basic components: the input device, the output device, the internal storage, the processor, and the auxiliary storage. Sometimes the internal storage and processor together are called the central processing unit or CPU.

Figure 1-1. The organization of a computer



* The word "data" is most correctly used as a plural noun. The singular of data is "datum." The usual practice, however, is to use the word data in a singular rather than plural sense. We will follow that practice in this book.

Figure 1-2. A mainframe computer. This is an IBM 3081. (Photo courtesy of International Business Machines Corporation.)



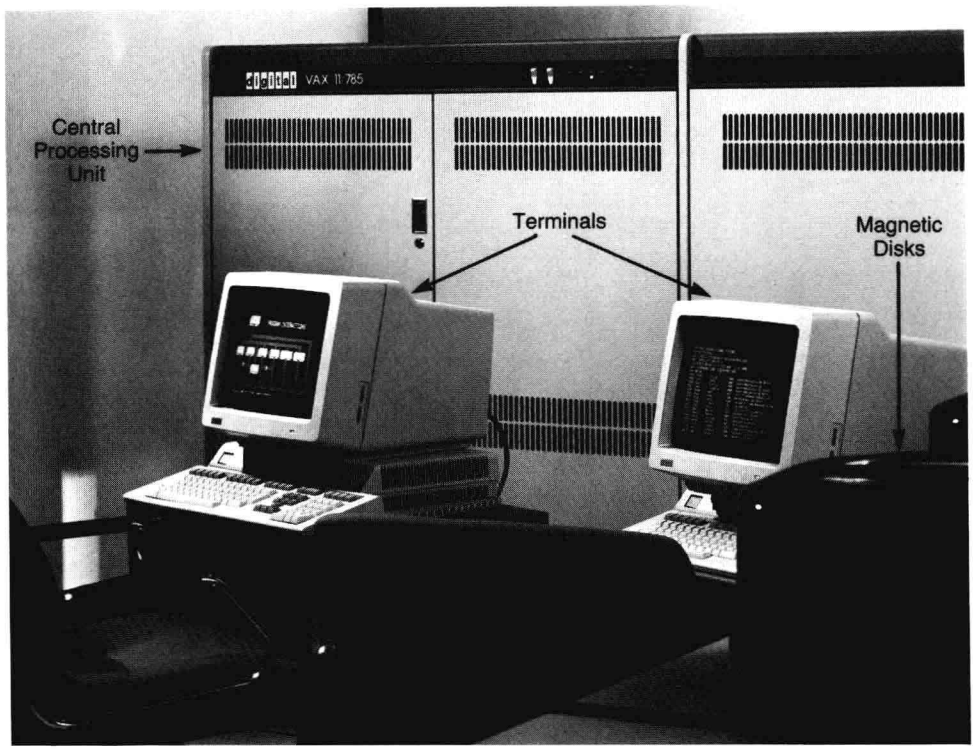
In this subsection we describe the components diagrammed in Figure 1-1. Figures 1-2, 1-3, and 1-4 show actual computers with the components discussed here.

Input and output devices. An *input device* is a mechanism that accepts data from outside the computer and converts it into an electronic form understandable to the computer. The data that is accepted is called *input data,* or simply *input.* For example, one common way to enter input into a computer is to type it with a typewriter-like *keyboard.* This keyboard is an input device. Each time a key is pressed, the electronic form of the symbol on the key is sent into the computer.

Another way to enter input into a computer is to use a device that reads the data from *punched cards* ("IBM" cards). Figure 1-5 shows an example of punched card input. The patterns of holes in the card represent different data. An input device for punched cards recognizes this data and transforms it into an electronic form understandable to the computer. Such a device is called a *card reader.*

An *output device* performs the opposite function of an input device. An output device converts data from its electronic form inside the computer

Figure 1-3. A minicomputer. This is a Digital VAX 11/785. (Photo courtesy of Digital Equipment Corporation.)



to a form that can be used outside. The converted data is called *output data,* or simply *output*. For example, one of the most common forms of output is a printed document or *report*. We often call this a computer "printout." Figure 1-6 shows an example of printed report output.

Printed output is produced by a device called a *printer,* which converts data from the computer into printed symbols to produce a paper copy of the output. Instead of being printed on paper, output is often displayed on a TV-like screen. Such a video display device is called a *CRT* for *c*athode *r*ay *t*ube (another name for a TV tube). When printer or CRT output is used with keyboard input, the devices are sometimes combined to form a unit called a *terminal.*

Input and output devices are often referred to together as input/output or *I/O* devices. Most computers have several I/O devices attached at one time. For example, a medium-sized computer may have many terminals and a printer. Some small computers, however, have only one input device and one output device (such as a keyboard and a CRT).