BASIC Programming for Chemists

An Introduction

Peter C. Jurs
Pennsylvania State University

Thomas L. Isenhour
Kansas State University

Charles L. Wilkins
University of California--Riverside

A Wiley-Interscience Publicati
JOHN WILEY & SONS
New York / Chichester

Library of Congress Cataloging in Publication Data:

Jurs, Peter C.
BASIC programming for chemists.

"A Wiley-Interscience publication."
Bibliography: p.
Includes index.
1. Chemistry—Data processing. 2. BASIC (Computer program language) I. Isenhour, Thomas L. II. Wilkins, Charles L. (Charles Lee), 1938- . III. Title.
QD39.3.E46J868 1987 005.13'3 87-13359
ISBN 0-471-85613-4

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

PREFACE

Computers have become an integral part of many aspects of modern life, and their use in chemistry is one outstanding example. As with other tools, computers affect chemistry in a variety of ways, some of which are obvious and some of which are hidden. Computers have been employed to perform many diverse duties, such as numerical calculations; data gathering; experiment modeling, simulation, and optimization; process and experiment control; information storage and retrieval; and graphic display of biological molecules for visual inspection. There is little doubt that computers will continue to become more and more important in determining how modern science will be conducted.

The present book has two objectives: to teach the fundamentals of the BASIC programming language by description and example; and to present a number of chemically oriented BASIC programs that can both teach about the language and be useful in their own right as well. Our aim has been to develop a text that would permit the reader to acquire the skill of computer programming from a chemist's viewpoint. For this reason, subject matter has not been segregated under headings such as programming language, numerical analysis, and chemical theory. Instead, the first three chapters present what we feel is necessary background for a beginning programmer in any field. Then, we immediately launch into the main part of the book, which consists of 53 example programs divided into 44 topics concerning chemical problems. These problems progress in difficulty in terms of their chemical concepts, mathematical models, and programming operations. The first problems and their accompanying programs are extremely simple, but the degree of difficulty increases, so that by the latter sections we are dealing with much more complex programs. By working these problems, studying the listings and flow diagrams, and by trying out the program, the student will rapidly become a competent programmer.

BASIC is a language ideally suited to this approach. Its interactive nature and student-oriented design ensure that the beginning programmer can proceed from simple examples to fairly complicated programming problems with a minimum of wasted effort. It has the additional advantage that the chemistry involved is not lost in a maze of complicated programming conventions. BASIC is an especially ideal vehicle for the person who may approach programming hesitantly, as a result of misconceptions about its difficulty.

The student should approach this book by reading through the first two chapters, "Introduction to Digital Computers," and "Computer Logic, Programming, and Flowcharts," and then follow this by studying thoroughly the main part of Chapter 3 on the BASIC language. Then the problems should be tackled, referring to the text and appendices as necessary. By all means, the programs should be copied and run and the results compared with the test examples. They should be modified, preferably with simple modifications at first, then with more complex ones, and tried out with these modifications. Later, the student should write original programs, progressing to complex programs at a self-paced speed. This book is built around the concept that we feel is certainly true: programming is the only way to learn to program.

PETER C. JURS
THOMAS L. ISENHOUR
CHARLES L. WILKINS

State College, Pennsylvania
Manhattan, Kansas
Riverside, California

CONTENTS

PART I

BASIC COMPUTER CONCEPTS AND BASIC

1

INTRODUCTION TO DIGITAL COMPUTERS

Computers are drastically altering our lives. In the modern world they enter into and affect virtually every aspect of people's activities. The ordering of automobile parts and butchers' aprons, the predictions of elections and of the weather, the scheduling of baseball games and cross-country buses, and even the gerrymandering of political districts are all problems that are routinely approached with computer analysis. Average individuals have their church contributions and life insurance billed by computer, their children's college curriculum and television sets designed by computer, and their paychecks and obituaries written by computer-generated directions.

Unfortunately, undesirable results from computer usage seem almost as frequent as desirable ones—to wit the phrases, "Your check will be 3 weeks late because the computer made a mistake," and "Official memo 1673—Grades must be submitted 3 days earlier this year because the University Central Records office has a new computer to increase efficiency."

The reason for the seemingly inconsistent effects of employing computers may be found by considering the computer as an intelligence amplifier. If stupidity is the negative of wisdom, then it follows that computerizing any process will only result in an amplification of the success or failure level to be expected of the programmer without a computer.

The computer is changing many aspects of the sciences, and chemistry is no exception from the general revolution that is now underway. To the modern chemist, a user's knowledge of the capabilities of the high-speed digital computer is rapidly becoming indispensible. Computers are used in a great variety of ways, but the major scientific uses may be summarized as follows:

1. Computers can routinely and conveniently perform numerical calcula-

tions more rapidly and accurately than if performed by hand. (A third generation computer of the IBM 370 class can perform approximately the equivalent of a person-life of calculations every 15 minutes, and more advanced computers are even faster.)

2. Computers can produce answers that would be useless if the time for hand calculations were required. (For example, the necessary calculations for correcting the trajectory of a spacecraft must usually be accomplished within a limited time. By the time a hand calculation could be done the answer might be only academic.)

3. Certain types of experiments may be optimized or simulated by computer calculations. (For example, knowledge of the functionality of the variable involved will often allow a computer simulation of a plant process. From such simulations the model may be optimized and the optimum plant constructed without a costly trial-and-error process of development.)

4. Computers are routinely used for information storage and the retrieval, organization, management, and presentation of large data banks. (Thousands of spectra may be searched and compared to that of an unknown, for example. The 10 nearest matches might then be presented. Such operations may be accomplished in a matter of seconds or minutes with a portion of the computer's various memory devices acting as a permanent file for all the library data.)

5. Computers can be used to gather data from experiments as produced (in real time), and they can often be incorporated into experimental apparatus so that they direct the experiment. For example, computerized X-ray diffractometers are sufficiently automatic to collect data for many days unattended.

6. Computers can be programmed to display intelligence by learning to perform tasks while improving their success as their experience increases. (Some advocates of machine intelligence go so far as to say that computers are essentially a new life form with nearly limitless possibilities.)

While the above list shows that computers are capable of a wide range of scientific activities, the emphasis in this book will be on the first category, numerical methods. This is because employing computers for performing calculations is the foremost use in beginning programming, and it is in many ways the easiest type of programming to discuss because of the concise nature of mathematical formulations. However, the emphasis on numerical examples and problems should not be construed to mean that computers are limited to such uses.

A modern general-purpose digital computer is an awesomely complicated electronic–mechanical assemblage. Anyone who has seen a large computer must admit to being impressed—at least on first encounter—with the variety of complicated events, including various flashing lights, spinning magnetic tapes, noisy high-speed printers and typewriters, clacking card readers, and the implication of thousands or even millions of hidden electronic circuits rapidly performing various complex operations.

Fortunately, one need not understand the internal operations of a computer to make good use of its capabilities. The computer may be treated in the same "black box" fashion that we treat many everyday mechanical and electrical devices. For example, an extremely complex apparatus used almost exclusively from a black box viewpoint is the ordinary television set.

Thus, it is apparent that a working knowledge of the internals of many devices is not required for adequate or even efficient utilization of these devices. However, the truly necessary knowledge is the relationship between the inputs and the consequent outputs of the device.

In keeping with this discussion, we hasten to point out that the majority of computer programmers, many of whom might be classed as experts, have little or no knowledge of the internal workings of a computer. Our approach to programming will assume that the reader has virtually no knowledge of the computer beyond the black box concept. Hence, we will deal exclusively with methods for obtaining desired responses, such as answers to calculations we wish to perform. However, for the interested reader, the next few pages of this introductory chapter give a brief, simplified look into the internal construction and organization of computers in general. It should be realized, however, that an understanding of this subject matter is in no way necessary for successful progress throughout the rest of the book.

INTERNAL ORGANIZATION OF COMPUTERS

The most fundamental property of a digital computer, indeed the source of the term *digital*, is that it operates by changing from one discrete state to another. We can compare this property to an electric lamp, the simplest of which has two states, OFF and ON. If we define OFF to represent the number 0 and ON to represent the number 1, then an electric lamp becomes a computer capable of adding 1 to 0 or subtracting 1 from 1. The lamp's input is in the form of a bistable switch, its memory is the position of the switch, and its output is the light (or lack of it) produced. Hence, if the lamp is in the OFF state, representing the number 0, 1 may be added to 0 by advancing the switch one step. On the other hand, if the lamp is in the ON state, representing the number 1, 1 may be subtracted from 1 by advancing the switch one position. (This step of the switch returns the lamp to OFF, the 0 position; hence the effect of subtraction has been accomplished.) Such a single binary device constitutes the simplest digital computer.

If we enlarge the machine by adding a second such stage it will have an increased capability. Now there are four possible different states: both OFF, 00 (binary 0); one OFF and the other ON, 01 (binary 1); one ON and the other OFF, 10 (binary 2); and both ON, 11 (binary 3). Thus, it is possible to carry out more extensive calculations and deal with larger numbers. This discussion may be extended to show that the number of discrete states of any binary digital system is equal to 2^n, where n is the number of individual binary components.

The process of addition in this binary system would be accomplished by adding one bit at a time to binary device *a*, and connecting the devices so that when device *a* is switched from ON to OFF, it adds 1 to device *b*. However, when device *a* switches from OFF to ON it would not affect device *b*. Counting would then proceed as follows:

	Device *a*	Device *b*	Binary Value	Decimal Value
	OFF	OFF	00	0
add 1	OFF	ON	01	1
add 1	ON	OFF	10	2
add 1	ON	ON	11	3

An analogy may be drawn between this system and the reels in an automobile odometer (mileage indicator). They function in the base 10 number system, but they are connected together in the same manner as the simple two-digit binary illustration above.

The working components of digital computers are largely electronic devices which deal with physically realizable voltages, currents, and magnetic fields. Many of these operations may readily be utilized in binary, or bistable, devices because of the phenomena upon which they are based. Hence, virtually all modern digital computers are internally binary machines. (This is not fundamentally necessary since currents, magnetic fields, etc., do have intensities and could be made to operate at several levels. However, it is normally much easier and quicker to detect the presence or absence of a current, or the orientation of a magnetic field, rather than their magnitudes. Furthermore, a very high degree of reliability can be realized in this manner.) Appendix A presents a discussion of binary arithmetic.

By combining many binary components, the computer is capable of assuming a large number of discrete states. These discrete states are actually the combination of the physical states of many thousand of electronic circuits or magnetic elements. The detailed discrete state of the computer at any particular time is usually not known, but it could be described fully in principle. This property is in contrast to continuous systems, which are not quantized and accordingly have an infinite number of states. (However, it must be realized that any real system has noise, bandpass, and sensitivity limitations which limit it to a finite number of distinguishable states.)

Digital computer components, therefore, assume discrete states that represent coded information in a binary form which symbolizes both numbers and the instructions for operating on them. These symbols are then manipulated internally to produce the desired calculations. *It is the extreme speed with which this is done, and the combinations of manipulations possible, that allow the digital computer to perform complex operations.*

The discrete nature of digital computers logically leads to some interesting and important consequences. The statement that computers are arithmetic does not mean they are merely fast adding machines. Computers are actually universal information processors. This universality is evidenced by the basic observation that any computer is capable of simulating any other computer. That is, any particular computer is a manifestation of an abstract universal information processor; as such it does not differ from other computers in abstract capacities, but only in practical ways. Much artificial intelligence research depends on the abstract information processing abilities of computers.

DIGITAL COMPUTER COMPONENTS

Figure 1.1 is a schematic diagram of the fundamental hardware components of a computer system. The central processing unit (CPU) performs arithmetic,

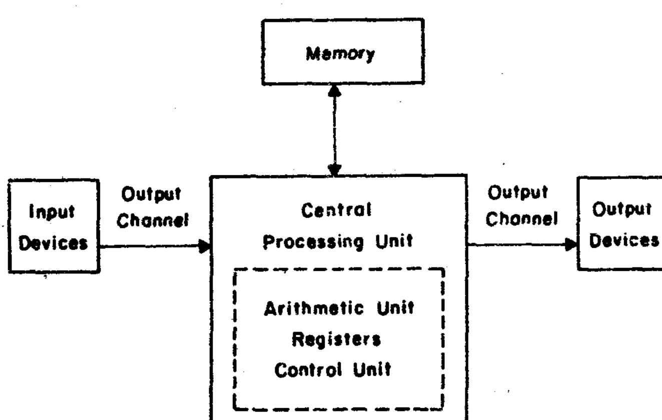

Figure 1.1. Schematic diagram of a computer system.

logical, and control operations. The main memory contains the current program and data. Input/output (I/O) devices provide communication between the peripherals and the computer. The peripherals include the operator's terminal, experiments, mass storage devices, fast printers, and other computers. The bus consists of the interconnections between each of the other components. Each of these is now discussed in more detail.

Central Processing Unit—CPU

The CPU directs and coordinates the overall functioning of the computer and contains the circuitry that actually executes the program instructions. The CPU has the following subunits:

1. *The arithmetic-logic unit*, which performs arithmetic or logical processing of data.
2. *Many registers* for short-term memory or storage of intermediate results of arithmetic instructions.
3. *The control unit*, which is able to address main storage in order to store or fetch information and to sequence instructions in the desired order.

The part of the computer that actually performs the arithmetic and logical operations is called the *arithmetic-logic unit*. It contains electronic circuitry that performs addition and subtraction in the binary number system and performs elementary logical operations. Larger computers additionally have circuitry that performs multiplication and division although small computers often perform these operations by executing simple programs involving repeated additions or subtractions. In actual practice, arithmetic or logical operations can be reduced to the two simple logical operations of (1) negation and (2) conjunction, which can both be conveniently implemented with electronic circuits.

The CPU contains several registers (readily accessible temporary storage units) which can be used to store (1) the numbers being handled by the arithmetic circuits of the machine, or (2) the address of the next instruction to be executed. An accumulator register (often called the *AC*) is used to store the sum or difference in addition or subtraction operations. A multiplier-quotient register (often called the MQ) is used along with the AC for multiplication and division. The MQ contains the multiplier or quotient and the AC stores the product or dividend, respectively, during multiplication or division.

The CPU also contains the control unit, which directs the sequence of operations to be performed by the computer. The various units of the computer are electronically linked by transmission lines which can be used to transmit information from one unit to another. The control unit is wired to respond to a repertory of basic instructions calling the other units into action as necessary. Typical instructions are as follows: Store the AC contents in a certain memory location, perform some operation on the contents of one of the registers, fetch the contents of a certain memory location, and so on. These instructions are executed by activating the transmission lines in the correct sequence by electronic logic circuits in the control unit. Thus, the control unit is the executive of the computer organization.

Main Memory

The main memory of the digital computer works in close conjunction with the CPU by storing and feeding back information to the CPU as instructed. This memory unit must be distinguished from storage on magnetic tape, disks, or other devices used for longer term storage. The memory is divided into sections each of which can store one word in a binary representation. A word may theoretically consist of any number of bits (binary digits) but is usually between 8 and 64 units, depending on the particular computer design. The term *word* is used because the quantity being stored can be a binary representation of a number, a letter or series of letters, a combination of letters and numbers, or a coded symbol for an instruction. The main storage not only stores numerical information such as data or intermediate results of a computation, but also stores the sequence of instructions that constitutes the program the computer is executing. Hence, the CPU calls on the main memory each time a stored instruction is to be executed.

Memories are usually classified by the number of memory cells they contain; large computer systems may have as many as tens of millions of memory cells. In order to be used by the CPU, each word in main memory has a unique address whose use allows the contents of that particular memory cell to be retrieved. Because the contents of any memory cell can be reached independently, the main memory is a random access memory (RAM). This feature is in contrast to magnetic tape, which is an example of a sequential, or serial, access device. Also, memory is classed as either read–write or read–only memory (ROM). The term RAM is generally used to describe read–write memory, which is either volatile or nonvolatile. Volatile memory loses its contents when power is removed, whereas nonvolatile memory is altered only by specifically writing into the memory cell. The majority of RAM in use today is volatile semiconductor memory. ROM is usually programmed by the manufacturer and is used when fast, permanent storage is necessary. In a general-purpose computer most of the memory is read–write since many different programs must be executed, which requires flexibility.

A measure of the speed and sophistication of a computer is the speed with which any word stored in the memory can be retrieved by the CPU for possible action. This is known as the *access time*. It varies from milliseconds for the earlier, slower computers, down to less than a microsecond for the faster, third generation computers of today. Of course, the access time of a computer has a direct bearing on the overall speed with which it can perform a calculation.

The minimum memory size required in a computer system is determined by the application, whereas the maximum size possible is limited by the hardware design. Eight bit microprocessors very often use two words (16 bits) to form an address, so they have a memory address range of 0 to $2^{16} - 1$, or 65535. Memory size is usually specified in bytes, where 1 byte is 8 bits. When referring to memory sizes a common symbol is k, for kilo. One k is 1024, so an 8-bit microprocessor, using 16-bit addressing, can address 64 kbytes of memory. Another common symbol is M, for mega, which is 1048576. The address range of computers with word sizes larger than 8 bits depends upon the specific implementation. Some larger computer systems can contain up to 64 Mbytes. Until the last few years, memory was very expensive, usually constituting a major portion of the cost of a computer system. Now, with high-density semiconductor RAM instead of core memory, up to 256 kbits may be stored upon a single integrated circuit. Memory costs are now only a small fraction of the total cost of a computer system. Small 8-bit personal computers now provide as much memory as the larger computers of only 10 years ago.

Input

There are many alternative methods for communicating with a modern digital computer. Input devices include cathode-ray tube (CRT) terminals, punched cards, magnetic tape, punched paper tape, magnetic disk, magnetic drum, electric typewriter, teletypewriter, and others. Each of these devices has its unique advantages and disadvantages, and the choice of which to use in a particular case depends on many circumstances. Table 1.1 lists the most common I/O devices along with their typical characteristics.

The usual I/O device on computers today is the CRT terminal. In the simplest version, it functions as an electronic typewriter. In more sophisticated versions, it can have dozens of special-purpose function keys.

The input medium most widely employed by computer users in the past was the punched card. Punched cards were produced on keypunches with keyboards similar to standard typewriters. Decks of punched cards could be convenient to handle, cards could be inserted or deleted easily, and they could be quickly processed by high-speed card readers.

Magnetic tape is commonly used where input information is repeatedly input in volumes too great to be easily handled by cards. Unlike cards, tapes are normally not produced directly by the user but are the result of some previous computer operation. One standard 2400-foot reel of one-half inch wide mag-