A Howard W Sams PHOTOFACT PUBLICATION • CMF-1

Fundamentals of
COMPUTER MATH

by ALLAN LYTEL • edited by A. A. WICKS

An introduction to the mathematics used in modern digital computer technology.

$4.95
CMF-1

FUNDAMENTALS OF COMPUTER MATH

by ALLAN LYTEL

CHAPTER 1. Numbers and the Computer—Introduces the concept of the radix, or base, of a numbering system. Discusses numbering systems having various bases and the sectional functions of a digital computer, as they relate to handling numerical data.

CHAPTER 2. Numbering Systems—Presents the conventional notation of the numbering systems important to digital computer technology. The binary, octal, and decimal systems are emphasized, and the methods of conversion between these systems are discussed.

CHAPTER 3. Numerical Operations—Explains coded systems, including the binary-coded decimal, and discusses the arithmetic operations pertaining to these systems.

CHAPTER 4. Polynomials and Linear Forms—Presents techniques for the evaluation of the roots of a polynomial, a short-cut method of evaluation of Taylor's expansions, and the methods of solution of simultaneous linear equations, both the method of elimination of variables and the method of solution by determinants.

(Continued on back flap)

(Continued from front flap)

CHAPTER 5. Interpolation—Introduces trigonometric functions as a prerequisite to the representation of a continuous function. The method of differences is given as a useful instrument for interpolation, and Newton's formulas for backward and forward interpolation are developed by using the method of differences.

CHAPTER 6. Other Techniques—Gives a short survey of various techniques in numerical analysis; the method of iteration for approximate solutions of simultaneous linear equations, the use of vectors and matrices, linear programming, and the essentials of relaxation methods.

CHAPTER 7. Other Mathematical Forms—Includes a brief discussion of symbolic logic and its application to switching circuits. Also presents various acronyms used in computer programming.

APPENDICES

Appendix I—A glossary of terms in common use in computer technology.

Appendix II — Fraçtional Decimal-to-Octal Conversion Tables.

Appendix III—Octal-to-Decimal Conversion Tables.

®

HOWARD W. SAMS & CO., INC.
THE BOBBS-MERRILL COMPANY, INC.

Fundamentals of COMPUTER MATH

by ALLAN LYTEL

The advantage of an electronic digital computer is its capability to perform mathematical operations at great speeds with maximum reliability. Properly guiding the computer in its work is the key problem in utilizing its capabilities.

The mathematical techniques associated with computer technology are covered in this text. The binary (two-valued) numbering system, whose values correspond to the on-off states of an electronic gate, is first developed so that number conversions between it and other systems can be made. Then it is possible to utilize the modern abstract mathematical techniques with a digital computer. Some of the methods covered in this text are iteration, linear programming, relaxations, numerical integration and differentiation, and interpolation. They are presented so as to allow for maximum latitude in the choice of techniques for problem-solving on the digital computer.

Written as a self-teaching guide for students and technicians, *Fundamentals of Computer Math* can also be used as a textbook in courses on modern digital computer technology, as well as a convenient reference for anyone involved with computer mathematics.

ABOUT THE AUTHOR

Allan Lytel, a graduate of both Temple and Syracuse Universities, holds a Master of Science degree in Journalism. He has taught mathematics and electronics at the Technical Institute of Temple University and radar maintenance in the Army Signal Corps. Author of more than a dozen books and innumerable articles, he has also prepared and conducted a course on digital computers. Mr. Lytel is currently manager of U.S. Electronics Publications, Inc. His other SAMS books include: *ABC's of Computers; ABC's of Computer Programming; ABC's of Boolean Algebra; Handbook of Electronic Charts & Nomographs; Handbook of Transistor Circuits; ABC's of Model Radio Control; Industrial X-Ray Handbook; Transistor Circuit Manual;* and *Automotive Electronics Test Equipment.*

ABOUT THE EDITOR

A. A. Wicks, a graduate of Sprott Radio College, is currently with the UNIVAC Division of the Sperry Rand Corp. He has conducted numerous radar operation studies for the National Research Council and has long been associated with publications in electronics.

HOWARD W. SAMS & CO., INC.
THE BOBBS-MERRILL COMPANY, INC.

$4.95
CMF-1

$4.95

Cat. No. CMF-1

Fundamentals of COMPUTER MATH

by ALLAN LYTEL

edited by A. A. WICKS

®

HOWARD W. SAMS & CO., INC.
THE BOBBS-MERRILL COMPANY, INC.
Indianapolis • New York

FIRST EDITION

FIRST PRINTING — FEBRUARY, 1964

FUNDAMENTALS OF COMPUTER MATH

Library of Congress Catalog Card Number: 63-23004

Fundamentals of COMPUTER MATH

Preface

A computer system can receive input information in many forms, perform arithmetical computations, and make logical decisions. Since the ultimate advantage of an electronic digital computer is its capability to perform simple mathematical operations at terrific speeds and with high reliability, the key problem in utilizing this power is in guiding the computer in its work. This is not, as might be expected, in knowing the correct controls to operate, but in analyzing a problem and preparing a list of instructions (programs) for the computer. The great amount of information that has been developed for computer programming has led to many programs and subprograms which have been thoroughly worked out and tested.

This book is concerned with the mathematical methods of providing instructions for the computer, and the mathematically based computer operations guided by these instructions. The first area discussed is that of how the computer handles the data. This leads naturally to a coverage of numbering systems and numbering representations, including the basic binary (two-valued) and octal (eight-valued) numbering systems. Then the basic mathematical operations within these systems are discussed and are subsequently incorporated into the mathematical techniques of problem-solving, such as the approximation of continuous functions, solutions of linear equations, and numerical integration and differentiation.

The subjects are categorized as follows: Numbers and the Computer; Numbering Systems; Numerical Operations; Polynomials and Linear Forms; Interpolation; Other Techniques, including simultaneous linear equations, use of vectors and matrices, and linear programming; and Other Mathematical Forms, including symbolic logic and its applications to switching circuits.

It is felt that this book will make it easier to solve the ever-growing problems of utilizing digital computers in data processing work. This content is designed to be used as either a self-teaching text for the technician and student or as a textbook for courses covering modern digital computer technology.

ALLAN LYTEL

January, 1964

Contents

CHAPTER 7

APPENDIX I

APPENDIX II

APPENDIX III

CHAPTER 1

Numbers and the Computer

The fundamental operations of a digital computer are exceedingly simple. As a system the computer can receive input information in many forms, perform arithmetical computations, and make logical decisions. The basic arithmetic operation of the computer is addition; subtraction is performed as an inverse of addition, and multiplication is performed as a series of additions. Division, by the same means, is performed as a series of subtractions.

INTRODUCTION

Since the ultimate advantage of an electronic digital computer is its capability to perform simple mathematical operations at terrific speeds and with high reliability, the key problem in utilizing this power is in guiding the computer in its work. This is not, as might be expected, in knowing the correct controls to operate (however, this is certainly extremely important), but in analyzing a problem and preparing a list of instructions for the computer. This function, programming, details a long series usually of fundamental and individual steps. (In contrast to this, the analog computer is more sophisticated, at least in terms of the mathematics that can be performed, since it is possible to integrate or differentiate quite easily in an analog device.)

Before considering the types of mathematics that can be accomplished on the digital computer, it is necessary to consider the special features as well as the limitations of the computer. In many cases, the digital computer is used where the input is a series of discrete individual counts or pulses and the problem is one of manipulation of these input data to perform the necessary computations. In other cases, where the input information is essentially analog, or a continuous function, the input data are necessarily broken up into a series of discrete pulses which are then used for the computation inside the machine. It is possible, of course, to graph the function by plotting the individual values and then connecting these individual values with a curve. This curve (Fig. 1-1) will closely approximate the function under discussion if a large number of points are plotted with great care. This, of course, is the method of aproximation, and there is extensive literature on the various techniques and methods of approximating functions for use in the digital computer.

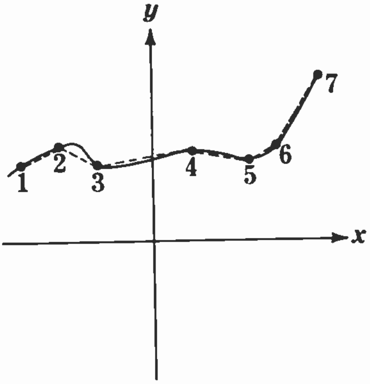


Fig. 1-1. Approximating a function.

COMMON NUMBERING SYSTEMS

The base of a number system is determined by the number of different digits used. Thus, in our common (Arabic) system of numbering, this is based on 10; hence, the decimal (deci = tenth) system.

We are most familiar with this system, but there are other, equally valid numbering systems using other bases. These include *biquinary*, using 10 by pairs, and the Roman system of numerals. Our decimal system is used for most desk calculators; the Roman numeral system is still used for building-cornerstone dates, and the biquinary system is the basis of the abacus and the IBM 650 computer.

The decimal system bears closer examination. Take the two digits 23, for example; in the decimal system we take this to mean three 1's and two 10's. To the base 5, however, this would mean decimal 13, because 23 is then three 1's and two 5's. Using the base 8, 23 would be decimal 19, for this is two 8's and three 1's.

This is shown in Table 1-1, where in the first column we count in tens, the second in fives, and the third in eights. Counting in decimal is 1, 2, 3, 4, 5, 6, 7, 8, 9, 10. To the base 5, the count is 1, 2, 3, 4, 10; and to the base 8, it is 1, 2, 3, 4, 5, 6, 7, 10.

The radix of a numbering system therefore being its base, the radix of the decimal system becomes 10. Because we use numbers so very often, we tend to forget that decimal numbers are not used in all applications. When we see 53 or VIII, what do we see? The number 53 means 3 more than 5 tens; VIII is *not* 53, but the eighth in the series of Roman numerals. It is one 5 and three 1's.

Let us examine this aspect further by using a trinary numbering system in which we count by A, B, C. As shown in Table 1-2, A is decimal 1, B is 2, while C is 3. Thus we count 1, 2, 3, or A, B, C. The next "number" is AA, which in decimal is one 3 plus one 1, or decimal 4.

In the same way:

$$\begin{array}{cccl}
(3^2) & (3^1) & (3^0) & \\
A & B & C & = 9+6+3=18 \\
B & A & C & = 18+3+3=24 \\
C & A & B & = 27+3+2=32 \\
C & C & B & = 27+9+2=38
\end{array}$$

This numbering system, as others, is an ordered set of digits used for counting. When all numbers, in sequence, have been used in the first position, the second position is occupied by the next symbol.

Thus our normal numbering system has 10 as its base. Any decimal number may be written as the sum of a power series

Table 1-1. Counting in Three Numbering Systems

	BASE	
10	5	8
1	1	1
2	2	2
3	3	3
4	4	4
5	10	5
6	11	6
7	12	7
8	13	10
9	14	11
10	20	12

of numbers, each of which is a power of 10. This may be seen by making columns of powers of tens and writing the digits in the proper place. The decimal numbers 352, 4,167, and 50,213 are written this way in Table 1-3. Each column, reading from right to left, has an increase of one in the power of 10, so that 4,167 is, for example, 4(1,000)+1(100)+6(10)+7(1). For mechanization, the decimal system of numbering (to the base ten) requires ten different and recognizable states for each digit. Any numbering system requires *n* states, where *n* is the numbering base.

Table 1-2. Trinary Numbering System

Decimal	Trinary	Decimal	Trinary
1	A	21	ACC
2	B	22	BAA
3	C	23	BAB
4	AA	24	BAC
5	AB	25	BBA
6	AC	26	BBB
7	BA	27	BBC
8	BB	28	BCA
9	BC	29	BCB
10	CA	30	BCC
11	CB	31	CAA
12	CC	32	CAB
13	AAA	33	CAC
14	AAB	34	CBA
15	AAC	35	CBB
16	ABA	36	CBC
17	ABB	37	CCA
18	ABC	38	CCB
19	ACA	39	CCC
20	ACB		

A computer built to accommodate the decimal system is certainly technically possible, but it would be unwieldy and expensive from a practical point of view. This is because the storing of figures in the decimal system would require ten different states, each needing to be easily identifiable and recoverable.

Any number can be expressed as a power series. However, any base can be used to express a given number; a base of 10 is normally used for ordinary computation. But, suppose the base is 5, as an example: a count from decimal 0 to 15 would be, to the base 5, as shown in Table 1-4. Representing the decimal number 2,469 to the base 5 would be easier as far as computer construction is concerned, requiring fewer states than the

Table 1-3. Decimal Notation

10^4	10^3	10^2	10^1	10^0
0	0	3	5	2
0	4	1	6	7
5	0	2	1	3

decimal system. If we write the base of a numbering system as the subscript of any number in that system, the preceding example becomes:

$$2{,}469_{10} = 34{,}334_5$$
$$34{,}334_5 = 3(5^4) + 4(5^3) + 3(5^2) + 3(5^1) + 4(5^0)$$
$$2{,}469_{10} = 1{,}875 + 500 + 75 + 15 + 4$$

Binary Numbering System

Electrically, there can be produced two basic states that may indicate "on" or "off," "go" or "stop," "white" or "black," or—and this is what permits a mathematical usage—"one" or "zero." Because of the practical considerations, then, digital computers use and represent numbers in the base 2, from which we derive the binary system.

The binary number system, to the base (or radix) 2, therefore, uses two characters: 0 and 1. The binary 1 and the binary 0 characters are referred to as *bits*, or *binary digits*. A binary 1 indicates the presence of a bit of information. A binary 0 indicates the absence of a bit of information. This does not mean

Table 1-4. Counting to the Base 5

Decimal	5^1	5^0
1	0	1
2	0	2
3	0	3
4	0	4
5	1	0
6	1	1
7	1	2
8	1	3
9	1	4
10	2	0
11	2	1
12	2	2
13	2	3
14	2	4
15	3	0

that the input to the computer must be in the binary system; it may be decimal or, for that matter, alphabetical or word-oriented. This is a matter of electronic conversion prior to processing by the computer. Other inputs, such as coded binary and coded decimal, may be used; these will be explained later.

Just as the first four powers of 10 are 1, 10, 100 and 1,000, the first four powers of 2 are 1, 2, 4, and 8, as in Table 1-5. Any quantity or magnitude which is expressed in the decimal system can be expressed in the binary system:

$$\begin{aligned} \text{Decimal:} \quad 19_{10} &= 1 \times 10^1 + 9 \times 10^0 \\ &= 10 + 9 \\ \text{Binary:} \quad 19_{10} &= 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \\ &= 16 + 0 + 0 + 2 + 1 \\ &= 1011_2 \end{aligned}$$

Note that for any integer n, 10^n is greater than 2^n. Thus, it is obvious that in the binary system many more positions are required to represent a given magnitude or quantity than are required in the decimal system.

Table 1-5. Powers of 10 and 2

System	Powers (Exponents)				
Decimal	1	10	100	1,000	10,000
Binary	1	2	4	8	16

COMPUTER PROGRAMMING

Computer programming is a technique and a field of its own, but a computer-user should know something of basic principles. Programming is the technique of setting up a specific series of steps by which a digital computer solves problems. To program a computer you must know the meaning of each individual step the computer can do. You must understand how to feed information into the machine, arrange the sequence of events, and let the machine solve your specific problem.

The individual and distinct functions of a digital computer are input, memory, arithmetic, control, and output.

Input devices read information into the computer from punched cards, punched tapes, or magnetic tapes. This information must first be translated into numbers, usually binary, to be read into the computer.

Information is read into some form of the computer *memory*. The input is either a set of instructions which tell the computer what to do, or information representing data for computation. Instructions, in total, form the computer program. Input data represent the information upon which the computer operates to perform its calculations.

The memory must be large enough to contain all of the necessary information and large enough that the information can be retrieved as fast as required. Magnetic cores are used in the large machines for the main memory, whereas magnetic drums, which are much slower, are used in some of the smaller machines. Magnetic tapes are also a form of computer memory.

The *arithmetic* section of the computer actually does the computations; it can add, subtract, multiply, divide, and make certain decisions. A register, which is often used as part of the arithmetic unit, is simply a place for the temporary storage of information. Information is placed in a register while it is being operated upon. Many computers have several registers where numbers are placed before, after, and during the various arithmetic operations.

Instructions are interpreted in the *control* section of the computer. In an "add" instruction, for example, the control section takes the instruction and internally arranges the circuits so that the computer will add two numbers.

The *output* of the computer takes information and stores it until it is needed. It also translates the information from computations to the necessary form, which may be printed sheets of paper, magnetic tapes, punched cards, or punched tapes. Computers can do several things simultaneously—read information from one input, provide information to several outputs, and at the same time compute another section of the problem.

Programming of a computer reduces to a coding problem after the problem has been defined in detail. Computers are of great value in solving problems that involve either a great many calculations or a large amount of data. Coded instructions are required for actual computer operation of a problem, and although the first electronic data-processing systems could be used only by experts, it now is possible to write machine-language programs that enable a computer to recognize instructions written in other languages; this led to the development of programming languages that were simpler to use than the machine language.

These "languages" are problem-oriented and permit the programmer to write convenient equivalents of machine instructions, using mnemonic symbols to represent them. The computer, acting under the control of previously written machine-language programs, then "translates" these instructions into equivalent machine instructions that are used in solving problems.

One such language is FORTRAN (*Mathematical FORmula TRANslating System*) developed by IBM. This language is written in algebraic and English notations; after the steps are written in this manner, the program is processed, by a computer, into a computer program.

CHAPTER 2

Numbering Systems

Numbering systems may be considered as being of two types: additive and positional. The Roman, Hebrew, Greek and Egyptian systems are additive. The decimal and binary systems are positional.

ADDITIVE NOTATION

An additive system is one in which symbols have different weights. In the Roman system, XVII means 17, whereas the Roman number XVII stands for 10 + 5 + 1 + 1. In this manner each symbol has a special, weighted value. Some symbols and their weights in this system are shown in Table 2-1.

Additive systems such as the Roman can be used for addition or subtraction without great difficulty, but they are extremely awkward for division and multiplication. To add 17 and 11, we have:

$$\begin{array}{l} \text{XVII} \\ \text{XI} \\ \hline \text{XXVIII} \end{array}$$

But, to multiply two numbers, such as MCXVII and MMCCI, is most difficult, although it can be done.

Table 2-1. Roman Numerals and Their Decimal Weights

Roman Symbol	Decimal Weight
I	1
V	5
X	10
L	50
C	100
D	500
M	1,000

POSITIONAL NOTATION

With the introduction of the concept of zero, the positional notation was developed. Note the absence of zero in the Roman system. In the positional notation, there is the concept of base, such as the base 10 in the decimal system. Other bases have been used in the past; consider angular measure with 60 seconds in a minute, 60 minutes in a degree, and 6 × 60, or 360, degrees in a circle.

Thus, in decimal, for the units' place we count from 0 to 9; for the next count, a 1 is used in the tens' place, while the units' place reverts to zero; thus 10 means $1 \times 10 + 0 \times 1$.

Any decimal number C can be expressed as:

$$C = A_n 10^n + A_{n-1} 10^{n-1} + \cdots + A_1 10^1 + A_0 10^0$$

For example: $156 = 1 \times 10^2 + 5 \times 10^1 + 6 \times 10^0$

$138 = 1 \times 10^2 + 3 \times 10^1 + 8 \times 10^0$

In general, for additive systems, any number N may be expressed as:

$$N = A_n X^n + A_{n-1} X^{n-1} + A_{n-2} X^{n-2} + \cdots + A_1 X^1 + A_0 X^0$$

where,

N is the number,
X is the base,
A is any digit that can be used.

In every case, the weight of a digit is X times the weight of the digit to its right. Thus, for base 10 each digit has 10 times the weight of the preceding (right-hand) digit. Table 2-2 shows the notations for bases 3, 4, 5, and 10. Thus for decimal 5, that number expressed in a system to the base is 012 $(1 \times 3 + 2 \times 1)$, to the base 4 is 011 $(1 \times 4 + 1 \times 1)$, and to the base 5 is 10 $(1 \times 5 + 0 \times 1)$. For decimal 12, that number expressed in a system to the base 3 is 110 $(1 \times 9 + 1 \times 3 + 0 \times 1)$, to the base 4 is 030 $(3 \times 4 + 0 \times 1)$, and to the base 5 is 22 $(2 \times 5 + 2 \times 1)$.

Computation (Table 2-3) follows from the structure of each positional notation for each base. Here are some examples for addition:

Decimal *Base 3*

$1 + 2 = 3 = 10_3 \; (1 \times 3^1 + 0 \times 3^0)$

$2 + 2 = 4 = 11_3 \; (1 \times 3^2 + 1 \times 3^0)$

Decimal *Base 4*

$$2 + 2 = 4 = 10_4 \ (1 \times 4^1 + 0 \times 4^0)$$

$$3 + 2 = 5 = 11_4 \ (1 \times 4^1 + 1 \times 4^0)$$

Decimal *Base 5*

$$3 + 3 = 6 = 11_5 \ (1 \times 5^1 + 1 \times 5^0)$$

$$4 + 3 = 7 = 12_5 \ (1 \times 5^1 + 2 \times 5^0)$$

Table 2-2. Positional Notation for Bases 3, 4, 5, and 10

Base 3	Base 4	Base 5	Base 10
000	000	00	00
001	001	01	01
002	002	02	02
010	003	03	03
011	010	04	04
012	011	10	05
020	012	11	06
021	013	12	07
022	020	13	08
100	021	14	09
101	022	20	10
102	023	21	11
110	030	22	12
111	031	23	13
112	032	24	14
120	033	30	15
121	100	31	16

For multiplication:

Decimal *Base 3*

$$2 \times 1 = 2 = 2_3 \ (2 \times 3^0)$$

$$2 \times 2 = 4 = 11_3 \ (1 \times 3^1 + 1 \times 3^0)$$

Decimal *Base 4*

$$2 \times 3 = 6 = 12_4 \ (1 \times 4^1 + 2 \times 4^0)$$

$$3 \times 3 = 9 = 21_4 \ (2 \times 4^1 + 1 \times 4^0)$$

Decimal *Base 5*

$$3 \times 3 = 9 = 14_5 \ (1 \times 5^1 + 4 \times 5^0)$$

$$4 \times 4 = 16 = 31_5 \ (3 \times 5^1 + 1 \times 5^0)$$

Table 2-3. Addition and Multiplication Tables From Positional Notation

ADDITION							MULTIPLICATION					
Augend / Addend	0	1	2			BASE	Multiplicand / Multiplier	0	1	2		
0	0	1	2			3	0	0	0	0		
1	1	2	10				1	0	1	2		
2	2	10	11				2	0	2	11		
Augend / Addend	0	1	2	3		BASE	Multiplicand / Multiplier	0	1	2	3	
0	0	1	2	3		4	0	0	0	0	0	
1	1	2	3	10			1	0	1	2	3	
2	2	3	10	11			2	0	2	10	12	
3	3	10	11	12			3	0	3	12	21	
Augend / Addend	0	1	2	3	4	BASE	Multiplicand / Multiplier	0	1	2	3	4
0	0	1	2	3	4	5	0	0	0	0	0	0
1	1	2	3	4	10		1	0	1	2	3	4
2	2	3	4	10	11		2	0	2	4	11	13
3	3	4	10	11	12		3	0	3	11	14	22
4	4	10	11	12	13		4	0	4	13	22	31

From these tables one could construct a complete set of tabular values for computation, if required.

Binary Notation

The value 2 is useful as a notational base. Switches for industrial control systems operate on a two-valued system, such as *on-off* or *open-closed* states. Vacuum tubes or transistors in modern digital computers also operate as simple switches which, as an example, either permit or prevent current flow. As stated in Chapter 1, these switching devices have only two

Table 2-4. Increasing Powers of 2

Powers of 2	←	2^4	2^3	2^2	2^1	2^0
Decimal Value	←	16	8	4	2	1

states and cannot easily be used for representing the decimal system.

The numbering system that is most easily developed for use in industrial control systems or in digital computers is therefore based on successive powers of 2. In this binary notation only two digits are used, 1 and 0. The first six powers of 2 are 1, 2, 4, 8, 16, and 32. Table 2-4 shows the increasing powers of 2 as 1, 2, 4, 8, and 16.

It is necessary that the base that is being used is known; this is in order to know what a quantity of 0's and 1's represents, since 10 represents 2 for a binary base and 10 for a decimal base; the number 3 expressed in the binary system is represented as 11, and it is *not* the decimal 11. Thus the decimals 1, 2, 3, 4, 5, and 6 correspond to binary 1, 10, 11, 100, 101, and 110. The binary number 101 is $4_{10} + 0_{10} + 1_{10}$, or decimal 5, written 5_{10}. In the same way 10111 is 23_{10}, or $16 + 0 + 4 + 2 + 1$. A table for binary numbers is developed in exactly the same way the powers of 10 were shown for the decimal system. From 1 to 10 decimal is (in binary) 01, 10, 11, 100, 101, 110, 111, 1000, 1001, 1010, as in Table 2-5.

Table 2-5. Decimal 1 Through 10 in Binary

Decimal	Binary			
	2^3	2^2	2^1	2^0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0

For the binary system the general expression for any number N is:

$$N_2 = A_n 2^n + A_{n-1} 2^{n-1} + \cdots + A_1 2^1 + A_0 2^0$$

Referring to Table 2-5, we have:

$$\begin{aligned} 010111_2 &= 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 \\ &\quad + 1 \times 2^1 + 1 \times 2^0 \\ &= 16 + 0 + 4 + 2 + 1 \\ &= 23_{10} \end{aligned}$$

The basic technique of expression is the same for binary as for decimal; 48_{10} means 4 tens plus 8 ones. In binary this is 110000, or one decimal 32 plus one decimal 16. Table 2-6 shows the binary representation of $2{,}469_{10}$; this is the sum of 2,048, 256, 128, 32, 4, and 1.

Binary numbers are powers of 2. Table 2-7 shows the values of positive and negative exponents of 2 from 0 to 500. Just as 10^{-1} is 0.1, so 2^{-1} is 0.5.

A negative exponent has this significance:

$$10^{-1} = \frac{1}{10^1} = \frac{1}{10} = 0.1$$

or:

$$10^{-2} = \frac{1}{10^2} = \frac{1}{100} = 0.01$$

For the binary system, this is:

$$2^{-1} = \frac{1}{2^1} = \frac{1}{2} = 0.5$$

or:

$$2^{-2} = \frac{1}{2^2} = \frac{1}{4} = 0.25$$

or:

$$2^{-3} = \frac{1}{2^3} = \frac{1}{8} = 0.125$$

To express a digit in the binary system or a bit only, a single indicator is needed in either the *on* (or 1 state) or *off* (which is the 0 state). Binary numbers are used for computation in a great many digital machines. The input data are converted to binary, usually in a coded system, the computations are made in the binary system, and the output data are converted from binary to the decimal system, for use external to the machine.

Binary Arithmetic

Binary numbers require very few rules and no tables for arithmetic operations. One advantage of the binary notation is the simplicity of operations.

Table 2-6. Binary Representation of $2{,}469_{10}$

2^{11} 2048	2^{10} 1024	2^9 512	2^8 256	2^7 128	2^6 64	2^5 32	2^4 16	2^3 8	2^2 4	2^1 2	2^0 1
1	0	0	1	1	0	1	0	0	1	0	1

Table 2-7. Powers of 2

n	2^n	2^{-n}
1	2	.50000 00000 x 10^{0}
2	4	.25000 00000 x 10^{0}
3	8	.12500 00000 x 10^{0}
4	16	.62500 00000 x 10^{-1}
5	32	.31250 00000 x 10^{-1}
6	64	.15625 00000 x 10^{-1}
7	128	.78125 00000 x 10^{-2}
8	256	.39062 50000 x 10^{-2}
9	512	.19531 25000 x 10^{-2}
10	1,024	.97656 25000 x 10^{-3}
11	2,048	.48828 12500 x 10^{-3}
12	4,096	.24414 06250 x 10^{-3}
13	8,192	.12207 03125 x 10^{-3}
14	16,384	.61035 15625 x 10^{-4}
15	32,768	.30517 57812 x 10^{-4}
16	65,536	.15258 78906 x 10^{-4}
17	131,072	.76293 94531 x 10^{-5}
18	262,144	.38146 97266 x 10^{-5}
19	524,288	.19073 48633 x 10^{-5}
20	1,048,576	.95367 43164 x 10^{-6}
21	2,097,152	.47683 71582 x 10^{-6}
22	4,194,304	.23841 85791 x 10^{-6}
23	8,388,608	.11920 92896 x 10^{-6}
24	16,777,216	.59604 64478 x 10^{-7}
25	33,554,432	.29802 32239 x 10^{-7}
26	67,108,864	.14901 16119 x 10^{-7}
27	134,217,728	.74505 80597 x 10^{-8}
28	268,435,456	.37252 90298 x 10^{-8}
29	536,870,912	.18626 45149 x 10^{-8}
30	1,073,741,824	.93132 25746 x 10^{-9}
35	3,435,973,837 x 10^{1}	.29103 83046 x 10^{-10}
40	1,099,511,628 x 10^{3}	.90949 47018 x 10^{-12}
45	3,518,437,209 x 10^{4}	.28421 70943 x 10^{-13}
50	1,125,899,907 x 10^{6}	.88817 84197 x 10^{-15}
60	1,152,921,505 x 10^{9}	.86736 17380 x 10^{-18}
70	1,180,591,621 x 10^{12}	.84703 29473 x 10^{-21}
80	1,208,925,820 x 10^{15}	.82718 06126 x 10^{-24}
90	1,237,940,039 x 10^{18}	.80779 35669 x 10^{-27}
100	1,267,650,600 x 10^{21}	.78886 09052 x 10^{-30}
125	4,253,529,586 x 10^{28}	.23509 88702 x 10^{-37}
150	1,427,247,693 x 10^{36}	.70064 92322 x 10^{-45}
175	4,789,048,565 x 10^{43}	.20880 97430 x 10^{-52}
200	1,606,938,044 x 10^{51}	.62230 15278 x 10^{-60}
300	2,037,035,976 x 10^{81}	.49090 93465 x 10^{-90}
400	2,582,249,878 x 10^{111}	.38725 91915 x 10^{-120}
500	3,273,390,608 x 10^{141}	.30549 36363 x 10^{-150}

Addition has three rules. These are illustrated below:

$$\begin{array}{r}0\\ \underline{0}\\ 0\end{array}\quad\begin{array}{r}0\\ \underline{1}\\ 1\end{array}\quad\text{or}\quad\begin{array}{r}1\\ \underline{0}\\ 0\end{array}\quad\begin{array}{r}1\\ \underline{1}\\ 10\end{array}$$

When these rules are followed, any two binary numbers may be added directly. For example, to add decimal 12 and decimal 5:

$$\begin{array}{r}1100\\ \underline{101}\\ 10001\end{array}$$

Also, for example, to add 01011 and 00110:

$$\begin{array}{r}01011\\ \underline{00110}\\ 10001\end{array}$$

Fractions (negative exponents of 2) are treated in the same way:

$$\begin{array}{rcl}10.11 &=& 2.75_{10}\\ \underline{1.01} &=& \underline{1.25_{10}}\\ 100.00 &=& 4.00_{10}\end{array}$$

For *multiplication* there are four rules which reduce to two: 1 times 1 is 1, and all other combinations are zero:

$$\begin{array}{r}1\\ \underline{\times 1}\\ 1\end{array}\qquad\begin{array}{r}0\\ \underline{\times 0}\\ 0\end{array}\qquad\begin{array}{r}1\\ \underline{\times 0}\\ 0\end{array}\qquad\begin{array}{r}0\\ \underline{\times 0}\\ 0\end{array}$$

Table 2-8. Binary Multiplication

		Multiplicand Digit	
	×	0	1
Multiplier Digit	0	0	0
	1	0	1

The binary multiplication table is shown in Table 2-8.

Because 0 and 1 are the only alternatives for the multiplier digit, either 0 or the multiplicand itself is used in forming the partial product. The partial products are added to provide the final product.

An example of binary multiplication would be 26 multiplied by 19:

$$\begin{array}{lr}
\qquad\qquad \textit{Decimal} & \textit{Binary} \\
26 = 16 + 8 + 0 + 2 + 0 = & 11010 \\
19 = 16 + 0 + 0 + 2 + 1 = & 10011 \\
\cline{2-2}
 & 11010 \\
 & 11010 \\
 & 00000 \\
 & 00000 \\
 & 11010 \\
\cline{2-2}
 & 111101110
\end{array}$$

Interpreting the binary result of the multiplication by using the ones, twos, fours system:

$$256 + 128 + 64 + 32 + 0 + 8 + 4 + 2 + 0 = 494$$

The product will be obtained by a series of adding the multiplicand and shifting whenever a 1 is found in the multiplier. For example, multiply 1011 by 0101:

$$\begin{array}{rl}
1011 & = 11 \\
0101 & = 5 \\
\cline{1-1}
1011 & \\
1011 & \\
\cline{1-1}
110111 & = 55
\end{array}$$

To multiply decimal 12 and 5:

$$\begin{array}{rl}
1100 & = 12 \\
101 & = 5 \\
\cline{1-1}
1100 & \\
1100 & \\
\cline{1-1}
111100 & = 60
\end{array}$$

With binary multiplication there are no carries (except in adding partial products), and every product is equal to 0 except for 1 × 1, which is 1. Multiplication is a series of additions, and multiplying 14 by 8 in decimal is the same as adding 14 eight times. In binary, 10101 can be multiplied by 101, as shown below:

$$\begin{array}{r}
10101 \\
101 \\
\hline
10101 \\
00000 \\
10101 \\
\hline
1101001
\end{array}$$

This multiplication is a series of shifts and additions. The 101 is a multiplier; as in the preceding example, a 1 is an addition and shift left; the 0 is a shift but no addition.

Shifting right or left is just a form of multiplication. Starting with 101100, a shift left is 1011000, or twice 101100. Dividing by 2 is a shift right, or 10110:

$$\begin{aligned} 1011000 &= 88 \\ 101100 &= 44 \\ 10110 &= 22 \end{aligned}$$

Table 2-9 is a tabular summary of addition and multiplication.

Division is the inverse of multiplication. In decimal division, we subtract the divisor from appropriate orders of the dividend, and to build the quotients, count the number of subtractions that do not produce a negative remainder. In binary division only one subtraction is necessary for each quotient digit. If it produces a positive remainder, then the quotient digit is 1. If it produces a negative remainder, the quotient digit is 0. The counting process is not required.

Consider the following example using the binary system to divide decimal 45 by decimal 9. Then we have 101101 ÷ 1001 = 101, or:

$$\begin{array}{r} 101 \\ 1001 \,\overline{)\,101101} \end{array}$$

			Quotient Digit
	101101		
Subtract	$\underline{100100}$		
	001001		1st = 1
Shift left	010010		
Subtract	$\underline{100100}$		
	101100	Negative	2nd = 0
Restore Add	$\underline{100100}$		
	010010		
Shift left	100100		
Subtract	$\underline{100100}$		
	000000		3rd = 1

Table 2-9. Binary Operations

Addition					Multiplication			
	+	0	1		×	0	1	
	0	0	1		0	0	0	
	1	1	10		1	0	1	

Two further examples are:

```
          110
  101 | 11110
        101
        ---
         101
         101
         ---
          00
```

and:

```
          101.11
  101 | 11101.00
        101
        -------
         1001
          101
          ----
          1000
           101
           ----
            110
            101
            ---
              1 Remainder
```

Table 2-10 shows a comparison of base 2 and bases 3, 4, 5, and 10 for addition and multiplication.

Table 2-10. Comparison of Bases for Addition and Multiplication

Decimal	Base 2	Base 3	Base 4	Base 5
Conversion				
54 23	110110 10111	2000 212	312 113	204 43
Addition				
54 (+)23 77	110110 10111 (carries) 1 1 1001101	2000 212 2212	312 113 1 431	204 43 11 302
Multiplication				
54 (+)23 162 108 1 (carry) 1242	1 1 0 1 1 0 1 0 1 1 1 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 0 0 0 0 0 0 1 1 0 1 1 0 1 1 1 10 10 1 1 1 (carries) 1 0 0 1 1 0 1 1 0 1 0	2000 212 1000 2000 11000 1 (carry) 1201000	312 113 2202 312 312 11 (carries) 103122	204 43 1122 1331 14432

Octal Notation

The octal number system is useful as a shorthand notation of the binary number system because eight is an integral power of two ($8 = 2^3$). One octal digit is always equal to three binary digits (and vice versa). Since a long string of ones and zeros cannot be effectively transmitted from one individual to another, some shorthand method is desirable.

The octal number system fills this need. Because of its simple relationship to binary, numbers can be converted from one system to another by inspection. The base or radix of the octal system is 8. This means there are eight symbols: 0, 1, 2, 3, 4, 5, 6, and 7. Notice that there are no 8's or 9's in this number system. The important relationship to remember is that three binary positions are equivalent to one octal position. Table 2-11 shows the equivalent of decimal numbers from 0 to 25 in binary and octal notation. This is a table that may be used when working in octal notation on a computer.

Table 2-11. Octal and Binary Equivalents

Decimal	Octal	Binary	Decimal	Octal	Binary
0	0	000	13	15	001101
1	1	001	14	16	001110
2	2	010	15	17	001111
3	3	011	16	20	010000
4	4	100	17	21	010001
5	5	101	18	22	010010
6	6	110	19	23	010011
7	7	111	20	24	010100
8	10	001000	21	25	010101
9	11	001001	22	26	010110
10	12	001010	23	27	010111
11	13	001011	24	30	011000
12	14	001100	25	31	011001

Beginning at the binary point, mark off the binary number into groups of three digits. Then replace each group with its octal equivalent. For example, convert 10110011 to decimal.

010	110	011	*Binary*
2	6	3	*Octal*
(2×8^2)	(6×8^1)	(3×8^0)	*Decimal*

Hence this is decimal 179, or octal 263.

A carry is also required in octal, for 001000_2 is 10_8 (since there is no 8 or 9 in octal). Thus:

$$\begin{aligned} 001001_2 &= 11_8 \\ 001010_2 &= 12_8 \\ 001011_2 &= 13_8 \\ 001100_2 &= 14_8 \end{aligned}$$

The octal numbering system to the base 8 is closely related to the binary system. Table 2-12 indicates the possibilities using octal addition. Thus, to add 7_8 and 1_8, we have:

$$\begin{array}{rcr} 7_8 & = & 111 \\ 1_8 & = & 001 \\ \hline 10_8 & = & 1000 \end{array}$$

Thus, in the octal system:

$$\begin{aligned} 1 + 1 &= 2 \\ 2 + 2 &= 4 \\ 3 + 3 &= 6 \\ 4 + 4 &= 10 \\ 5 + 5 &= 12 \\ 6 + 6 &= 14 \\ 7 + 7 &= 16 \end{aligned}$$

Binary, octal, and decimal notations may be compared as in the following:

$$\begin{array}{rrr} \textit{Binary} & \textit{Decimal} & \textit{Octal} \\ 010 & 2 & 2 \\ \underline{111} & \underline{7} & \underline{7} \\ 1001 & 9 & 11 \\ & & \\ 101 & 5 & 5 \\ \underline{110} & \underline{6} & \underline{6} \\ 1011 & 11 & 13 \end{array}$$

For example, to add 321_8 and 405_8 we have:

$$\begin{array}{ccc} 3 & 2 & 1 \\ 4 & 0 & 5 \\ \hline 7 & 2 & 6 \end{array}$$

Table 2-12. Octal Addition

+	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	10
2	2	3	4	5	6	7	10	11
3	3	4	5	6	7	10	11	12
4	4	5	6	7	10	11	12	13
5	5	6	7	10	11	12	13	14
6	6	7	10	11	12	13	14	15
7	7	10	11	12	13	14	15	16

Or, to add 767 and 654 in the octal system, we have:

$$\begin{array}{rrr} 7 & 6 & 7 \\ 6 & 5 & 4 \\ \hline 16 & 4 & 3 \end{array}$$

This octal system can be used on special desk calculators. Adding 321$_8$ and 405$_8$ in binary would be as follows:

$$\begin{array}{ccccccc} 011 & 010 & 001 & = & 3 & 2 & 1 \\ 100 & 000 & 101 & = & 4 & 0 & 5 \\ \hline 111 & 010 & 110 & = & 7 & 2 & 6 \end{array}$$

Notice that the translation from binary to octal is quite simple, as 111 is 7$_8$, 010 is 2$_8$, and 110 is 6$_8$.

Table 2-13 shows octal multiplication. Perhaps an example of this can best be seen by steps: consider 457 multiplied by 263:

$$\begin{array}{rcrcr} 457 & \quad & 457 & \quad & 457 \\ 3 & & 6 & & 2 \\ \hline 25 & & 52 & & 16 \\ 17 & & 36 & & 12 \\ 14 & & 30 & & 10 \\ \hline 1615 & & 3432 & & 1136 \end{array}$$

Table 2-13. Octal Multiplication

×	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7
2	0	2	4	6	10	12	14	16
3	0	3	6	11	14	17	22	25
4	0	4	10	14	20	24	30	34
5	0	5	12	17	24	31	36	43
6	0	6	14	22	30	36	44	52
7	0	7	16	25	34	43	52	61

By addition of these partial products we have:

```
  1615
 3432
1136
------
151735
```

Thus, while the digital computer itself uses the binary system of numbering, the octal system can be used by a human operator (with or without a desk calculator) to check on the operation of the computer during the testing of a new program. Conversion from the binary (computer language) to octal system, which can be used by the human operator, is direct and simple. Conversion from decimal to binary is the last remaining step needed for going in any direction between decimal, binary, and octal.

NUMBER CONVERSIONS

For simplification, numbering conversions are frequently made, including binary, octal and decimal. There are five different number conversions in use:

1. Decimal to Octal
2. Octal to Decimal
3. Octal to Binary (Binary to Octal)
4. Binary to Decimal
5. Decimal to Binary

Decimal-to-Octal Conversion

Decimal integers can be converted to octal by successively dividing the integer by 8, and noting the remainders. The remainders are ignored in the following divisions, and the divisions are continued until the decimal integer is reduced to zero. The remainders are then combined to form the octal equivalent. Here is an example:

```
              R
8 | 159       7     ↑
   8 | 19     3     |
      8 | 2   2     |
          0         |
```

Read the answer *up* as 237_8.

The rules are as follows:

1. Divide the original decimal number by 8.

2. This remainder (7) becomes the low order (least significant) digit.
3. The quotient of the first division (19) is divided again by 8.
4. This remainder (2) is the next order digit.
5. When the quotient (2) is less than the divisor, this quotient becomes the highest order digit.

Thus in this case the answer is 237_8, which is equal to 159_{10}.

Decimal fractions can be converted to octal fractions by means of successive multiplications by 8. The integers produced by the multiplication become the octal digits. When the multiplications are continued, the integers are ignored. The multiplications are performed until the fraction becomes zero. Here is an example in converting 0.149_{10} to octal:

$$\begin{array}{c|r}
I & \\
 & .149 \\
 & \underline{8} \\
1 & .192 \\
 & \underline{8} \\
1 & .536 \\
 & \underline{8} \\
4 & .288 \\
 & \underline{8} \\
2 & .304
\end{array}$$

Thus the octal number is .1142+, which is binary .001001100010.

The steps are:

1. Multiply .149 by 8 to get 1.192; consider 1 as the first significant digit.
2. Multiply the .192 by 8 to get 1.536; consider the 1 as the next significant digit.
3. Multiply .536 by 8 to get 4.288; consider 4 as the next digit.
4. In the same way, the last digit is 2.

Table 2-14 shows decimal values from 10^0 to 10^{10} and their octal conversions. Here are some examples:

$$10^0 = 001_8 = 001_2$$
$$10^2 = 144_8 = 001100100_2$$
$$= 100_{10}$$

Table 2-15 shows negative decimal to octal values. Table 2-16 gives some constants that often appear in calculations.

If a decimal fraction cannot be represented exactly as an octal fraction, the multiplication process will never produce a result of 0. In this case, multiplication is performed until a satisfactory approximation is reached. This approximation limit is determined by the computer programmer at the time the program is developed. When the computer reaches this point it will do one of two things: it will stop, or it will continue with the next step.

Octal-to-Decimal Conversion

Octal integers can be converted to decimal integers by multiplying each octal digit by the correct power of 8. The lowest order digit is multiplied by 8^0, the next higher order digit by 8^1, the next by 8^2, and so on. When the products are added together, the equivalent decimal integer is obtained. For example, consider 143_8:

Table 2-14. Decimal Values and Octal Conversions

DECIMAL	OCTAL
10^0	+ 000000 000001
10^1	+ 000000 000012
10^2	+ 000000 000144
10^3	+ 000000 001750
10^4	+ 000000 023420
10^5	+ 000000 303240
10^6	+ 000003 641100
10^7	+ 000046 113200
10^8	+ 000575 360400
10^9	+ 007346 545000
10^{10}	+ 112402 762000

$$\begin{aligned} 143_8 &= 1(8^2) + 4(8^1) + 3(8^0) \\ &= 1(64) + 4(8) + 3(1) \\ &= 64 + 32 + 3 \\ &= 99_{10} \end{aligned}$$

There is a short cut, which is the following series of steps:

1. Multiply the highest order digit by 8.
2. Add the next highest order digit.
3. Multiply this sum by 8.
4. The process is completed when the lowest order digit has been added.

In the preceding example of 143_8:

$$\begin{array}{rl} 1 & \text{the highest order digit (1)} \\ \underline{8} & \\ 8 & \text{multiplied by 8.} \\ \underline{4} & \\ 12 & \text{the next highest order digit (4) is added.} \\ \underline{8} & \\ 96 & \text{the sum (12) is multiplied by 8.} \\ \underline{3} & \\ 99 & \text{the final, or lowest order digit (3) is added.} \end{array}$$

$$143_8 = 99_{10}$$

For octal numbers less than 1_8, the highest order digit (i.e., the digit nearest the point) is multiplied by 8^{-1}, the next highest order by 8^{-2}, the next by 8^{-3}, and so on. When the products are added together, the equivalent decimal fraction is obtained. As an example, consider $.1142_8$:

$$\begin{aligned} .1142_8 &= 1(8^{-1}) + 1(8^{-2}) + 4(8^{-3}) + 2(8^{-4}) \\ &= \frac{1}{8} + \frac{1}{64} + \frac{4}{512} + \frac{2}{4096} \\ &= \frac{610}{4096} \\ &= .1489_{10} \end{aligned}$$

Conversions are given in Table 2-17 for $\frac{1}{n}$ in octal and decimal for the range of n from 2 to 30. From Table 2-17, we can see that:

$$\begin{aligned} 0.5_{10} &= 0.4_8 \\ 0.25_{10} &= 0.2_8 \\ 0.125_{10} &= 0.1_8 \end{aligned}$$

For assistance in conversions several appendices are included. Appendix III overs integers for octal-to-decimal conversions from 00000 octal (which is 00000 decimal). Appendix II covers fractional decimal-to-octal conversions for decimal values from 0.000 to 0.999; it provides octal values with factors of 1, 10^{-3}, and 10^{-6}.

Binary–Octal Conversion

This process is performed almost by inspection and is quite simple. The binary number (for binary to octal) is divided into groups of three bits, starting from the right, and then

Table 2-15. Negative Powers of 10 (Octal)

10^{-1}	.06314	63146	31463	14631	46314	63146
10^{-2}	.00507	53412	17270	24365	60507	53412
10^{-3}	.00040	61115	64570	65176	76355	44264
10^{-4}	.00003	21556	13530	70414	54512	75170
10^{-5}	.00000	24761	32610	70664	36041	06077
10^{-6}	.00000	02061	57364	05536	66151	55323
10^{-7}	.00000	00153	27745	15274	53644	12742
10^{-8}	.00000	00012	57143	56106	04303	47375
10^{-9}	.00000	00001	04560	27640	46655	12263
10^{-10}	.00000	00000	06676	33766	35367	55653
10^{-11}	.00000	00000	00537	65777	02745	44453
10^{-12}	.00000	00000	00043	13631	40226	75204
10^{-13}	.00000	00000	00003	41134	11502	22732
10^{-14}	.00000	00000	00000	26411	15606	50226
10^{-15}	.00000	00000	00000	02200	72763	67165
10^{-16}	.00000	00000	00000	00163	22545	13731
10^{-17}	.00000	00000	00000	00013	41675	24311
10^{-18}	.00000	00000	00000	00001	11622	73507
10^{-19}	.00000	00000	00000	00000	07301	71124
10^{-20}	.00000	00000	00000	00000	00571	62410
10^{-21}	.00000	00000	00000	00000	00045	61664
10^{-22}	.00000	00000	00000	00000	00003	61622
10^{-23}	.00000	00000	00000	00000	00000	30133
10^{-24}	.00000	00000	00000	00000	00000	02326
10^{-25}	.00000	00000	00000	00000	00000	00174
10^{-26}	.00000	00000	00000	00000	00000	00014
10^{-27}	.00000	00000	00000	00000	00000	00001

each three-bit number is converted into its octal equivalent. For example, the binary number 101010 is converted to its octal equivalent as follows:

$$\begin{array}{cc} 1\ 0\ 1 & 0\ 1\ 0 \\ (2^2 + 1) & (2^1) \\ = 5 & = 2 \end{array}$$

$$= 52 \text{ (octal)}$$

Table 2-16. Octal Representation for Common Constants

Miscellaneous Constants	Decimal	Octal
$\sqrt{2}$	1. 414 213 562 4	1. 324 047 463 201
$\sqrt{3}$	1. 732 050 807 6	1. 566 636 564 132
$\sqrt{5}$	2. 236 067 977 5	2. 170 673 633 460
$\sqrt{6}$	2. 449 489 742 8	2. 346 107 024 023
$\sqrt{7}$	2. 645 751 311 1	2. 512 477 651 650
$\sqrt{8}$	2. 828 427 124 8	2. 650 117 146 402
$\sqrt{10}$	3. 162 277 660 2	3. 123 054 072 667
π	3. 141 592 653 6	3. 110 375 524 211
2π	6. 283 185 307 1	6. 220 773 250 413
$1/\ \pi$	. 318 309 886 2	. 242 763 015 564
$1/2\pi$	. 159 154 943 1	. 121 371 406 672
1° = 1/360 of a circle	. 002 777 777 8	. 001 330 133 015
e	2. 7 8 281 828 5	2. 557 605 213 053
1/e	. 367 879 441 2	. 274 265 306 615
$\log_{10}e$	. 434 294 481 9	. 336 267 542 512
$\log_e 10$	2. 302 585 093 0	2. 232 730 673 533
$\log_e 2$	. 693 147 180 6	. 542 710 277 600
$\log_{10}\pi$	. 497 149 872 7	. 376 424 666 307
$\log_e\pi$	1. 144 729 885 8	1. 112 064 044 344

Octal-to-binary conversions are made by simply reversing the process. Each digit of the octal number is replaced by its three-bit binary equivalent. Thus, the octal number 725 is converted to its binary equivalent:

$$\begin{array}{ccc} 7 & 2 & 5 \\ (111) & (010) & (101) \end{array}$$

$$= 111010101 \text{ (binary)}$$

Fractions are treated in the same way:

$$.001 \quad 001 \quad 100 \quad 010 \quad = \quad .1142_8$$

Binary-to-Decimal Conversion

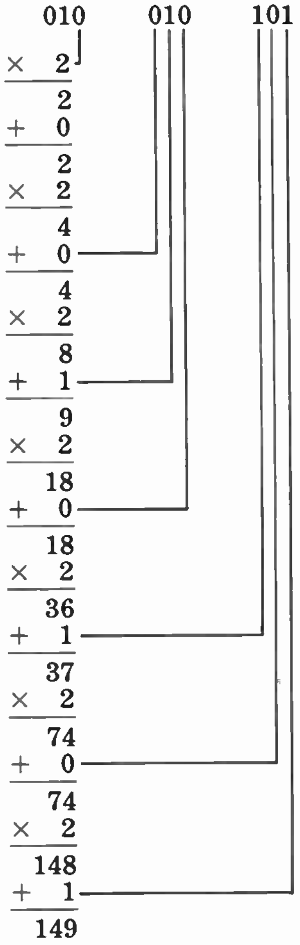
There are two methods for this conversion. In one, binary integers can be converted to decimal numbers by means of successive multiplications and additions. First, the highest order bit is multiplied by 2. Then the product is added to the next lower order bit and this sum multiplied by 2. The new product is then added to the next lower order bit and the re-

Table 2-17. Octal-to-Decimal Conversion

n	1/n (octal)	1/n (decimal)
2	.400000000000	.5000000000
3	.252525252525	.3333333333
4	.200000000000	.2500000000
5	.146314631463	.2000000000
6	.125252525252	.1666666667
7	.111111111111	.1428571428
8	.100000000000	.1250000000
9	.070707070707	.1111111111
10	.063146314631	.1000000000
11	.056427213505	.0909090909
12	.052525252525	.0833333333
13	.047304730473	.0769230769
14	.044444444444	.0714285714
15	.042104210421	.0666666667
16	.040000000000	.0625000000
17	.036074170360	.0588235294
18	.034343434343	.0555555555
19	.032745032745	.0526315789
20	.031463146314	.0500000000
21	.030303030303	.0476190476
22	.027213505642	.0454545454
23	.026205441310	.0434782609
24	.025252525252	.0416666667
25	.024365605075	.0400000000
26	.023542354235	.0384615384
27	.022755022755	.0370370370
28	.022222222222	.0357142857
29	.021517345410	.0344827586
30	.021042104210	.0333333333

sultant sum multiplied by 2. This multiplication and addition is continued bit by bit to the right until the lowest order bit is added. No multiplication follows the adding of the lowest order bit because this is in the units column. The resulting sum is the decimal equivalent of the binary integer. An example is given below:

Convert 010010101 to its decimal equivalent:



OR, 10010101

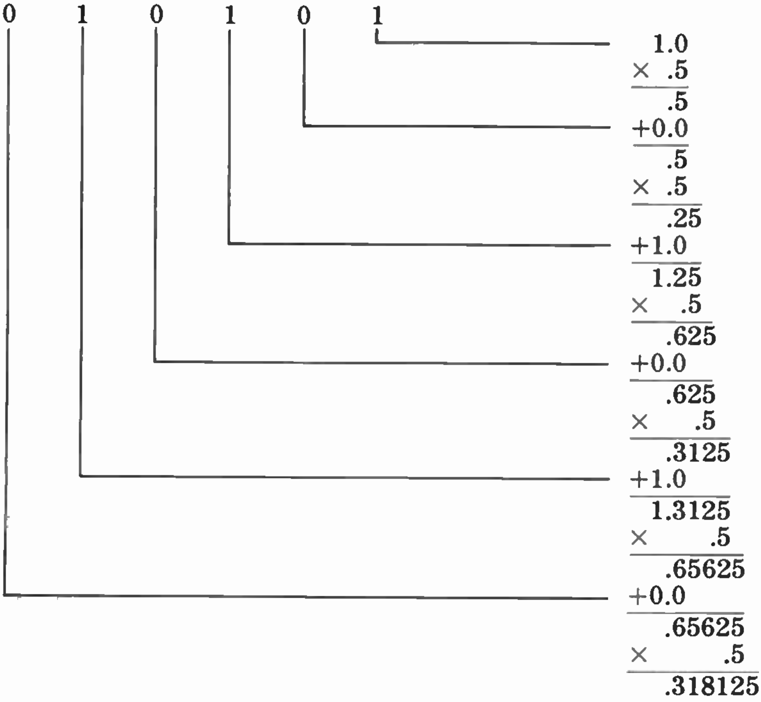
$$= 1(2^7) + 0(2^6) + 0(2^5) + 1(2^4) + 0(2^3) + 1(2^2) + 0(2^1) + 1(2^0)$$

$$= 128 + 16 + 4 + 1$$

$$= 149_{10}$$

Binary fractions can be converted to decimal fractions by means of successive multiplications by 0.5 and by the performance of bit additions.

An example for the conversion of .010101 is shown below:



```
0     1     0     1     0     1
                                      1.0
                                    ×  .5
                                       .5
                                     +0.0
                                       .5
                                    ×  .5
                                       .25
                                     +1.0
                                      1.25
                                    ×   .5
                                       .625
                                     +0.0
                                       .625
                                    ×    .5
                                       .3125
                                     +1.0
                                      1.3125
                                    ×      .5
                                       .65625
                                     +0.0
                                       .65625
                                    ×       .5
                                       .318125
```

During the conversions, however, each bit is reduced to its correct power of 2 because of the successive multiplications by 0.5. The conversion is begun at the extreme right-hand bit. This bit is multiplied by 0.5, and the resultant product is added to the next higher-order bit. This sum is then multiplied by 0.5, and the new product is added to the next higher-order bit. This multiplication and the addition is continued bit by bit to the left until the highest-order fraction bit is added. This digit is then multiplied by 0.5. The resultant sum is the decimal equivalent of the binary fraction.

Fractions may also be converted as in the following example for .001001100010:

$$
\begin{aligned}
.001001100010 &= 1(2^{-3}) + 1(2^{-6}) + 1(2^{-7}) + 1(2^{-11}) \\
&= \frac{1}{8} + \frac{1}{64} + \frac{1}{128} + \frac{1}{2048} \\
&= \frac{305}{2048} \\
&= .1489_{10}
\end{aligned}
$$

Double-Dabble—This is a system for binary-to-decimal conversion. You start with a digit; if the next is a 0, you double what you have; if the next is a 1, you dabble (double plus 1). For example, consider 10101:

1. Start with the MSD (most significant digit), which is equal to 1.
2. Next is 0; double 1, which is 2.
3. Next is 1; dabble 2, which is 5.
4. Next is 0; double 5, which is 10.
5. Next is 1; dabble 10 which is 21.

Thus $10101_2 = 21_{10}$.

Decimal-to-Binary Conversion

Decimal integers can be converted to binary numbers by successively dividing the integer by 2. After each division, the 0 or 1 remainder is recorded. In succeeding divisions the remainders are ignored and the divisions are continued until zero is obtained. The remainders are the resultant binary number when they are combined in an order such that the first remainder is the lowest-order binary digit, the second remainder the next higher-order digit, etc.

For example, convert 270_{10} to its binary equivalent:

```
2|270          Remainder 0   ↑  Read
 2|135         Remainder 1   |  Up
  2|67         Remainder 1   |
   2|33        Remainder 1   |
    2|16       Remainder 0   |
     2|8       Remainder 0   |
      2|4      Remainder 0   |
       2|2     Remainder 0   |
        2|1    Remainder 1   |
           0
```

Thus, $270_{10} = 100001110_2$.

Decimal fractions can be converted to binary fractions by successively multiplying the fraction by 2. A 0 is recorded each time the product remains smaller than 1; a 1 is recorded each time the product exceeds 1. The integer portion of the result is dropped when the multiplication is continued. This multiplication process is continued until the answer becomes exactly 1. The bits are then combined to form the binary fraction. For example, consider $.149_{10}$:

$$\begin{array}{c c|r}
\text{Read} & & .149 \\
\text{Down} & & \times\ 2 \\ \cline{3-3}
\Bigg\downarrow & 0 & .298 \\
& & \times\ 2 \\ \cline{3-3}
& 0 & .596 \\
& & \times\ 2 \\ \cline{3-3}
& 1 & .192 \\
& & \times\ 2 \\ \cline{3-3}
& 0 & .384 \\
& & \times\ 2 \\ \cline{3-3}
& 0 & .768 \\
& & \times\ 2 \\ \cline{3-3}
& 1 & .536 \\
& & \times\ 2 \\ \cline{3-3}
& 1 & .072 \\
& & \times\ 2 \\ \cline{3-3}
& 0 & .144 \\
& & \times\ 2 \\ \cline{3-3}
& 0 & .288 \\
& & \times\ 2 \\ \cline{3-3}
& 0 & .576 \\
& & \times\ 2 \\ \cline{3-3}
& 1 & .152 \\
& & \times\ 2 \\ \cline{3-3}
\downarrow & 0 & .304
\end{array}$$

Thus, $.149_{10} = .001\ 001\ 100\ 010\ +$

CHAPTER 3

Numerical Operations

As we have seen, digital computers use binary digits during manipulation in the computer to produce the basic operations of addition, subtraction, multiplication, and division. The representation we have discussed thus far has been "pure binary." That is, the ones and zeros making up a number have a place-value that represents some power of two. However, in many instances the pure binary system is not practical for computers to implement.

CODED SYSTEMS

Obviously, a pure binary system would require the manipulation of large digital quantities, both in and out of the computer. For instance, the decimal number 147 expressed in pure binary is 10010011. Various systems have therefore been devised to reduce this manipulation of large notations. These systems are essentially coded systems, and we have within these binary-coded decimal codes the excess-three code (shortened to XS-3), the 8421 and 7421 codes, and the Gray code. There are also special error-detecting codes and others which have been devised.

Binary-Coded Decimal (BCD) Systems

Let us consider the coding of the preceding example, 147. If we represent each individual decimal digit in binary but keep their decimal values regarding position, we have:

0001	0100	0111
1	4	7

With computers operating in coded decimal, the information can be placed in or removed from the computer directly. But if

Table 3-1. Decimal, Binary, and BCD Notations

Decimal Notation	Binary Notation	Binary-Coded Decimal Notation	
00	00000	0000	0000
01	00001	0000	0001
02	00010	0000	0010
03	00011	0000	0011
04	00100	0000	0100
05	00101	0000	0101
06	00110	0000	0110
07	00111	0000	0111
08	01000	0000	1000
09	01001	0000	1001
10	01010	0001	0000
11	01011	0001	0001
12	01100	0001	0010
13	01101	0001	0011
14	01110	0001	0100
15	01111	0001	0101
16	10000	0001	0110
17	10001	0001	0111
18	10010	0001	1000
19	10011	0001	1001
20	10100	0010	0000

we were using pure binary, this would have to be converted to decimal form. Although there are a number of inefficiencies in using them, coded forms are very easily converted by the computer operator or programmer.

A *weighted code* is one in which each binary digit contributes a specific decimal value to the number. A *non-weighted code* is a code in which a constant decimal value cannot be assigned to each binary digit position. In the first case, the 8421 code is a weighted code because each digit has a decimal value of 8, 4, 2 or 1. An example of a non-weighted code is the XS-3 code, which will be described.

In Table 3-1, each decimal digit is given a binary-coded decimal representation (the binary notation is also shown). You will note that zero is represented as all zeros. This is a disadvantage in either pure binary or BCD, since there is no clear-cut definition of the difference between true zero and a failure.

XS-3 Code—In the XS-3 code each number is represented by a binary number that is three more than its actual value.

Table 3-2. Binary to XS-3 Code

BINARY	XS-3	
00	0011	0011
01	0011	0100
02	0011	0101
03	0011	0110
04	0011	0111
05	0011	1000
06	0011	1001
07	0011	1010
08	0011	1011
09	0011	1100
10	0100	0011
11	0100	0100
12	0100	0101
13	0100	0110
14	0100	0111
15	0100	1000
16	0100	1001
17	0100	1010
18	0100	1011
19	0100	1100
20	0101	0011

This permits direct subtraction using complements, which will be described later. In the XS-3 code, binary 0011 (3_{10}) is added to each number to produce the code. Table 3-2 shows this code. To subtract 3, which is 0110, from 1100, or 9, the 0110 is changed to 1001, which is 6. In the same manner, 2 or 0101 from 9 is 7 or 1010. This is a simplification of computer arithmetic.

The 7421 Code—In addition to the XS-3 code already mentioned, there is the 7421 code in which the fourth digit represents 7 rather than 8. This simple change, it turns out, never

Table 3-3. Relationship of 7421 and 8421 Codes

Decimal	7421 Code	8421 Code
0	0000	0000
1	0001	0001
2	0010	0010
3	0011	0011
4	0100	0100
5	0101	0101
6	0110	0110
7	1000	0111
8	1001	1000
9	1010	1001

requires more than two ones for any number up to 9. Thus it requires less power consumption than pure binary. Table 3-3 shows this code.

Biquinary Code—This code has been used in a number of computers, notably the IBM 650. It has an advantage in providing easy error detection. Each valid digit has only two binary ones, with one bit in the "bi" portion and one in the "quinary" part. Therefore all single- and triple-bit errors in each part are detected, as well as many double-bit errors. In the IBM 650, the biquinary code is converted to a five-level code for storage. Table 3-4 shows this code.

Table 3-4. Decimal to Biquinary Code

Decimal	Binary	Biquinary
0	0000	01 00001
1	0001	01 00010
2	0010	01 00100
3	0011	01 01000
4	0100	01 10000
5	0101	10 00001
6	0110	10 00010
7	0111	10 00100
8	1000	10 01000
9	1001	10 10000

Gray Code—This code, also known as the *cyclic code*, or *reflective code*, is used where a conversion of a shaft rotation (analog data) is being converted to digital data. The chief advantage of this code is that only a single binary digit is changed in passing from the representation of one decimal number to the next greater or lesser number.

In the design of analog-to-digital converters it is convenient to change one digit at a time as the shaft rotates, and for this reason the Gray code is widely used for equipment such as shaft encoders or decoders.

In pure binary there are times when only one digit changes, as from 2 to 3 in decimal, which is from 0010 to 0011 in binary; or from 8 to 9 in decimal, which is from 1000 to 1001 in binary. However, there are many other cases where more than one digit changes in binary, as from 9 to 10 (decimal), which is 1001 to 1010; or 1011 to 1100, from 10 to 11. Because of this the Gray code is used, since in this code there is always a change of just one digit in going from one number to the next, higher or lower. Table 3-5 shows the decimal, pure binary, and Gray codes.

By inspection, conversion from binary to the Gray code may be deduced to follow these rules:

1. The binary number is added to itself (shifted or indexed one place to the right).
2. The right-most digit is dropped.
3. All carries are ignored.

For example, 9_{10} is 1001_2. For the Gray code, we have:

$$\begin{array}{r l} 1001 & \\ \underline{1001} & \text{(indexed left)} \\ 1001x & \end{array}$$

where x is the dropped digit.

Thus for decimal 9 the Gray code representation is 1101. In the same way for 11_{10}, we have:

$$\begin{array}{r} 1011 \\ \underline{1011} \\ 1110x \end{array}$$

And so 1110 in the Gray code represents 11 in decimal.

Table 3-5. Decimal to Gray Code

Decimal	Binary	Gray Code
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010

Error-Detecting Codes—In general, computers manufactured today provide automatic means to prevent equipment-caused errors from continuing, by stopping the computation, an alarm, or both. However, errors can occur in the control portion of the computer or in the input, and a means must be provided to detect these errors.

One method of error-checking has already been noted: use of the biquinary code. Another, which is very commonly used, is the *parity* (or *odd-even*) *code*. In this method an extra binary digit, called the check digit, is used for each character code. This bit is assigned so that the total number of 1's is always odd. For example, in the case of 0001 nothing is added, and it appears as 0001. For 0011 (3_{10}) a 1 is added, so it appears as 10011 with the first bit being the parity bit.

Thus, by checking each number for an odd number of bits (1, 3, 5, 7, and so on), a check for errors is obtained to prevent dropping a 1.

Table 3-6. Two-out-of-Five Error-Detection Code

Decimal	Two-out-of-Five Code
0	00011
1	00110
2	01100
3	11000
4	10001
5	10010
6	10100
7	01010
8	00101
9	01001

For the same technique with even-parity, 0001 becomes 10001, 0011 is unchanged, 0100 becomes 10100, and so on.

It is also possible to use a two-out-of-five code in which only two of the five digits are 1's, which is a further error check. One type of two-out-of-five code is shown in Table 3-6.

ARITHMETIC OPERATIONS

The ordinary arithmetic operations (addition, subtraction, division, and multiplication) can be straightforward with decimal and with binary numbers. But there are some special cases used by computers. Some of these are: *subtraction by complements; binary complements; fixed radix-point numbers; floating radix-point numbers;* and *shifting.*

Shorthand Decimal Operations

We can examine briefly some of the basic operations with decimal numbers to see how these operations may be approached in different ways.

Division offers several possibilities if it is considered as repeated subtraction. Consider 157 divided by 12. By long division this is:

$$\begin{array}{r} 13 \\ 12\overline{)157} \\ \underline{12} \\ 37 \\ \underline{36} \\ 1 \end{array}$$

But, as repeated subtraction, we have:

$$\begin{array}{rcl} 157 & & \\ \underline{120} & 1 & (12 \times 10) \\ 37 & & \\ \underline{12} & & \\ 25 & 1 & (12 \times 1) \\ \underline{12} & & \\ 13 & 1 & (12 \times 1) \\ \underline{12} & & \\ 1 & = & \text{Remainder} \end{array}$$

As the first step the divisor (12) is multiplied by 10 (or 100 or 1,000 as the case may be) and subtracted from the dividend. Note that 1,200 would be too large to give a positive difference. The first digit of the quotient is 1, and, since 12 was written as 120, the 1 appears as 10.

Next, the 12 is subtracted from 37 as many times as is necessary to obtain a remainder less than 12. In this case, 12 is subtracted 3 times, with a remainder of 1. Thus, the quotient is 10 + 3, or 13, with a remainder of 1.

Consider 10,763 divided by 53; this yields 203 + 4R.

$$\begin{array}{rcl} 10763 & & \\ \underline{5300} & 1 & (53 \times 100) \\ 5463 & & \\ \underline{5300} & 1 & (53 \times 100) \\ 163 & & \\ \underline{53} & 1 & (53 \times 1)^{*} \\ 110 & & \\ \underline{53} & 1 & (53 \times 1) \\ 57 & & \\ \underline{53} & 1 & (53 \times 1) \\ 4 & R & \end{array}$$

* Note that 53 × 10, or 530, is too large; hence, it is not used.

In summation, we obtain:

$$\begin{aligned} 2(53 \times 100) + 3(53 \times 1) + 4R &= 200 + 3 + 4R \\ &= 203 + 4R \end{aligned}$$

Multiplication can also be shortened by using the right-hand digits and left-hand digits with no carries.

Multiplying 97 by 83 in the usual manner gives 8,051. Using the left-hand digits only (such as 2 for 21, 3 for 37, or 4 for 46, for example) one obtains:

$$\begin{array}{rl} 97 & \text{(LHD)} \\ \underline{83} & \\ 22 & \\ \underline{75} & \\ 772 & \end{array}$$

For right-hand digits:

$$\begin{array}{rl} 97 & \text{(RHD)} \\ \underline{83} & \\ 71 & \\ \underline{26} & \\ 331 & \end{array}$$

The final sum is:

$$\begin{array}{r} 331 \\ \underline{772} \\ 8051 \end{array}$$

This product can be written so that all carries are deferred until the final step:

$$\begin{array}{rl} 97 & \\ \underline{83} & \\ 220 & \\ 7500 & \text{LHD} \\ 71 & \\ \underline{260} & \text{RHD} \\ 8051 & \end{array}$$

Consider a further example. In the normal manner of multiplication, we have:

$$\begin{array}{r} 437 \\ \underline{752} \\ 874 \\ 2185 \\ \underline{3059} \\ 328{,}624 \end{array}$$

By left- and right-hand digits, we have:

```
      RHD           LHD

      437           437
      752           752
      ---           ---
      864           001
     055           213
    819           224
    -----         -----
    83314         24531
```

Adding the partial products, we obtain:

```
      83314
     24531
     -------
     328,624
```

COMPLEMENTS

The arithmetic operations of a computer are forms of addition and subtraction, and this function is enhanced by making it possible for the computer to process these operations by using complement arithmetic. Complement arithmetic was mentioned in the discussion on XS-3 codes and is that form of arithmetic that makes it possible to perform subtraction by means of addition, or addition by means of subtraction.

The *complement* of a number is that quantity which, when added to the number, produces a power of the base of the number, or a power of the base minus one. The former is called the *ten's complement* in binary arithmetic. The latter is called the *nine's complement* in decimal arithmetic and the *one's complement* in binary arithmetic. This is more fully explained later.

Counting

Counting is a basic arithmetic operation and is also another way of viewing addition. A counter with two digits starts from 00 and progresses to 01 as the first count, through 99 for a full count, after which it returns to 00. There are 100 possible numbers, hence it has a modulus 100, or 10^2. Thus a counter with modulus 10^5 would count to 99,999, or 10^5-1.

Consider a cyclic counter as in Fig. 3-1 where positive (clockwise) counting, starting from 0, goes in unit steps until 100 is reached, which is the point 0 again. A count of 25 is one-quarter around the circle, 50 is half-way, and so on. Negative counting is done counterclockwise. Thus, on this counter to subtract 35 from 50 (add a negative 35 and positive 50) one would go clockwise 50 units, then counterclockwise from this point 35 units to stop at plus 15. From this, one can see that the outer numbers (positive) and the inner numbers (negative) are complements. For, in every case, we have:

$$| n_1 | + | n_2 | = M$$

where,

n_1 is any inner number,
n_2 is the corresponding outer number,
M is the modulus.

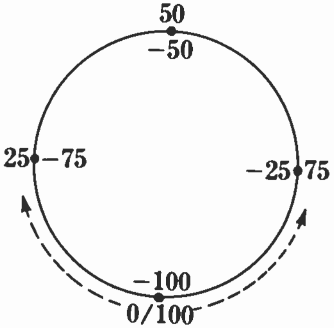


Fig. 3-1. Operation of a cyclic counter.

Counting is related to addition, because addition is an extension of counting where the last possible number is infinity. There is always a finite last number in any usable system, and it depends on the number of columns in the counting device. Consider a counter which counts the 100,000 numbers from 00,000 to 99,999; after 99,999 the counter goes back to 00,000. This counter has a modulus of 100,000 or 10^5, where 5 is the number of columns in the counter.

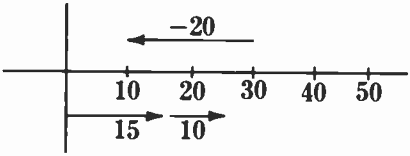


Fig. 3-2. Counting represented linearly.

Counting can be performed in two directions as in Fig. 3-2. If counting to the right is counting forward, counting to the left is counting backward. A count to the right means an addition, a count to the left is subtraction.

A practical counter with a fixed modulus is represented by a circle in which counting counterclockwise may be called forward counting and counting clockwise may be called backward counting. Positive numbers, or forward counting, and the negative numbers, or backward counting, can be placed side by side around the circle. This was shown in Fig. 3-1 where the corresponding numbers are "complements" of each other.

The value of the complement concept is very significant in the performance of subtraction by a mechanized system where subtraction may be performed by reversing the direction of movement of a wheel. By converting negative numbers to their complements it is possible to convert all problems into addition problems for which the wheel always turns in the same direction.

When the modulus is given in the decimal system, the complement is called the ten's complement, and it is defined as that number obtained when each digit is subtracted from 9, and a 1 added to the result. Here is an example showing the ten's complement of 456:

$$\begin{array}{rr} & 999 \\ - & 456 \\ \hline & 543 \\ + & 1 \\ \hline & 544 \end{array}$$

In the binary system there is a two's complement, which is defined as that number obtained when each bit is subtracted from 1 and a 1 added to the result. Here is an example deriving the two's complement of 1010:

$$\begin{array}{rr} & 1111 \\ - & 1010 \\ \hline & 0101 \\ + & 1 \\ \hline & 0110 \end{array}$$

The number obtained by subtracting each digit from 9 is called the nine's complement, and the number obtained by subtracting each bit from 1 is called the one's complement (there is no addition of 1, as is the case for the two's complement). The nine's complement of 456 is therefore 543; and the one's complement of 1010 is 0101. Note that in the binary system, obtaining the one's complement is very easy; an interchange of 0's and 1's is all that is required.

Subtraction By Complements

A complement is a useful quantity for operations such as subtraction. The *true complement* of any quantity in positional notation is that quantity which, when added to the first quantity, gives the least quantity containing one more place. The *base-minus-one's complement* of any quantity is the quantity which, when added to the first quantity, gives the largest quantity containing the same number of places.

For the decimal system the true complement of a one-digit number m is $(10 - m)$. Thus, subtracting 427 from 564:

$$\begin{array}{r} 564 \\ -\ 427 \\ \hline 137 \end{array}$$

The ten's or true complement of 427 is 573, since we have:

$$\begin{array}{r} 573 \\ 427 \\ \hline 1{,}000 \end{array}$$

From the previous definition, 573 when added to 427 gives the smallest number (1) in the fourth place.

Thus, subtracting 427 from 564, we have:

$$\begin{array}{r} 564 \\ -427 \\ \hline 137 \end{array}$$

Complementing 427 and adding to 564 gives:

$$\begin{array}{r} 564 \\ 573 \\ \hline 1137 \end{array}$$

The MSD, or extra 1, is dropped (since the answer is just 1,000 too large).

Consider another example:

$$\begin{array}{r} 392 \\ -141 \\ \hline 251 \end{array} \quad \text{or,} \quad \begin{array}{r} 392 \\ +859 \\ \hline 1{,}251 \end{array} \text{ (which is 251)}$$

The nine's complement is the difference between each digit and 9. Thus, for 451 the nine's complement is 548. Subtracting 451 from 863 yields:

$$\begin{array}{r} 863 \\ -451 \\ \hline 412 \end{array}$$

But, by using nine's complement, we obtain:

$$\begin{array}{r} 863 \\ +548 \\ \hline 1{,}411 \\ \hookrightarrow 1 \\ \hline 412 \end{array}$$

The "extra" 1 is an end-around carry to complete the sum. Table 3-7 shows the decimal complements. Consider now an example using the ten's complement: Subtracting 427 from 564 yields:

$$\begin{array}{r} 564 \\ -427 \\ \hline 137 \end{array}$$

Table 3-7. Decimal Complements

	Decimal	
Digit	True Complement	Nine's Complement
0	10	9
1	9	8
2	8	7
3	7	6
4	6	5
5	5	4
6	4	3
7	3	2
8	2	1
9	1	0

The ten's complement of 427 is 573; and adding this to 564:

$$\begin{array}{r} 564 \\ +573 \\ \hline 1{,}137 \end{array}$$

Thus, the answer is 137, by dropping the extra first digit. The nine's complement of 427 is 572, or just 1 less than the true ten's complement (573). Adding again:

$$\begin{array}{r} 564 \\ +572 \\ \hline 1{,}136 \end{array}$$

But here the extra 1 is added to the 136, which is the sum without the first 1 digit:

$$\begin{array}{r} 136 \\ +1 \\ \hline 137 \end{array}$$

Thus, subtraction is somewhat more complicated than addition; it involves "borrowing" from the next higher column whenever a subtrahend digit is greater than the corresponding minuend digit. But it is possible to obtain the same result by adding the minuend to the complement of the subtrahend and ignoring the final carry.

Note that 4 can be subtracted from 7 in either of the following manners.

By normal subtraction:

$$\begin{array}{r} 0007 \\ -0004 \\ \hline 0003 \end{array}$$

Adding the true complement gives:

$$\begin{array}{r} 0007 \\ +9996 \\ \hline 10{,}003 \end{array}$$

Note that all places, including nonsignificant digits, of the subtrahend must be complemented, and that the resulting sum always exceeds the capacity of the register. When the true complement of the subtrahend is used, the carry from the highest order of the register is ignored. Because the base-minus-one's complement is more easily obtained, subtraction is often accomplished by adding the minuend to the nine's complement of the subtrahend, and performing an end-around carry from the highest order column to the lowest order column. For example, to subtract 4 from 7, add the nine's complement and 1:

$$\begin{array}{r} 0007 \\ +9995 \\ \hline 0002 \\ \llcorner\!\!\longrightarrow 1 \\ \hline 0003 \end{array}$$

Binary Complements—In binary, the nine's complement of 1011 is 0100; this is obtained by interchanging the 1's and 0's. To subtract 1011 from 101101 directly:

$$\begin{array}{r} 101101 \\ -001011 \\ \hline 100010 \end{array}$$

And since the complement of 001011 is 110100, subtraction is also possible by addition:

$$\begin{array}{rl} 101101 & \\ +110100 & \\ \hline 1100001 & \\ +\qquad 1 & \text{(end-around carry)} \\ \hline 100010 & \end{array}$$

Another example is subtracting 0101 from 1011:

$$\begin{array}{r} 1011 \\ -0101 \\ \hline 0110 \end{array}$$

or,

$$\begin{array}{r} 1011 \\ +1010 \\ \hline 10101 \\ +1 \\ \hline 0110 \end{array}$$

Table 3-8 shows the complements of the binary-coded decimal.

Table 3-8. Complements of BCD Numbers

Decimal	Binary-Coded Decimal		
	Digit	True Complement	Nine's Complement
0	0000	1010	1001
1	0001	1001	1000
2	0010	1000	0111
3	0011	0111	0110
4	0100	0110	0101
5	0101	0101	0100
6	0110	0100	0011
7	0111	0011	0010
8	1000	0010	0001
9	1001	0001	0000

Some codes have the advantage of being self-complementing. Each binary digit of any decimal representation is the complement of the corresponding binary digit in the representation of the decimal complement. One of these self-complementing codes is the previously mentioned XS-3 code, in which each decimal digit is represented by the binary equivalent of the digit plus three.

The XS-3 code with its complements is shown in Table 3-9. In XS-3 coding, the binary representations of each decimal place are added in binary fashion. Carries from one decimal place to the next are determined by carries from the highest-order binary place. Because the decimal digits are each expressed in XS-3 code, the sum of two digits will be expressed in XS-6 code, which can be converted back to XS-3 code by subtracting three. Table 3-10 shows the complement pairs of the XS-3 code.

Table 3-9. XS-3 Code and Nine's Complement

Decimal Digit	Excess-Three	Nine's Complement
0	0011	1100
1	0100	1011
2	0101	1010
3	0110	1001
4	0111	1000
5	1000	0111
6	1001	0110
7	1010	0101
8	1011	0100
9	1100	0011

RADIX-POINT OPERATIONS

The significance of the radix must not be lost in numerical operations. In manual arithmetic, the decimal point locates the units' order of a number, and separates the negative and positive powers of the base. For example, in the decimal number 724.32, the quantity to the left of the decimal point equals $(7 \times 10^2) + (2 \times 10^1) + (4 \times 10^0)$; and the right-hand portion: $(3 \times 10^{-1}) + (2 \times 10^{-2})$. This sharp definition is not required for computer use, however, because the power coefficients are placed in a predetermined position, and all operations are maintained on this basis. This is known as *fixed-point representation* and is described under the next heading. It is customary in this type of operation to place the decimal point to the left of the most significant digit, so that *all* numbers in the computer manipulation are considered to have values of less than 1.

In all probability such an arrangement would limit the range of numbers to be used in a computer. Therefore, in program-

Table 3-10. Complement Pairs of XS-3 Code

Decimal	Binary	XS-3
0	0000	0011 ←
1	0001	0100 ←
2	0010	0101 ←
3	0011	0110 ←
4	0100	0111 ←
5	0101	1000 ←
6	0110	1001 ←
7	0111	1010 ←
8	1000	1011 ←
9	1001	1100 ←

ming a problem for the computer, the programmer must (after initially determining the location he will use for the decimal point) position all numbers in correct relationship to the position of the decimal point selected. As long as each step or operation is complete within itself, the decimal-point position can be changed as the program progresses.

The step whereby the decimal point of a number is located so as to be within the capability of a computer is known as *scaling*, which is essentially varying the units or scale that defines the number. Given a decimal number of, say, 500.0, we may also express it as 5×10^2. If we move the position of the decimal point, we also move the position of the power of the base. Therefore, we could also state that the 5 in 50.0 is equal to 5×10^1. In scaling, the digits of the number are moved in direct relationship to some fixed but assumed position of the decimal point. This is the same as multiplying by a power of the base, which will be equal to the same number of positions that the digits have been shifted. If the shift of the decimal point is made to the right of its normal position, this parallels a multiplication of a positive power of the base; if it is to the left, it is the same as a multiplication by a negative power of the base.

Fixed-Point Numbers

In fixed-point operation the point (decimal point or binary point) is established for a particular run or program on a computer as a convenience. Suppose that you were operating with these numbers:

$$
\begin{aligned}
A &= 13{,}740.49\\
B &= 5{,}067.78\\
C &= 1{,}548.90\\
D &= 330.54
\end{aligned}
$$

It is possible to work with these, as has been mentioned, by dividing each by 100 and instead use:

$$
\begin{aligned}
A &= 137.4049\\
B &= 50.6778\\
C &= 15.4890\\
D &= 3.3054
\end{aligned}
$$

You must, however, recognize the problems this creates. Take A = 100.0 and B = 200.0 (two three-digit numbers). We add them and get a three-digit number:

$$\begin{array}{r} 100.0 \\ +200.0 \\ \hline 300.0 \end{array}$$

Or we subtract them and also get a three-digit number:

$$\begin{array}{r} 100.0 \\ -200.0 \\ \hline -100.0 \end{array}$$

But, notice that when we multiply we no longer obtain a three-digit number:

$$\begin{array}{r} 100.0 \\ \times 200.0 \\ \hline 20{,}000.0 \end{array}$$

The product is at least a five-digit number.

If we had operated with A as 1.0 and B as 2.0, by dividing 100 and 200 each by 100, then the multiplication would have been:

$$\begin{array}{r} 1.0 \\ \times 2.0 \\ \hline 2.0 \end{array}$$

This product (2.0) cannot be multiplied by 100, it must be multiplied by 100 × 100, or 10,000, to get the proper answer. Thus:

$$\begin{array}{lrcr} 1.0 \times & 100 & = & 100 \\ 2.0 \times & 100 & = & 200 \\ \hline 2.0 \times & 10{,}000 & = & 20{,}000 \end{array}$$

This, of course, comes about from the following rule:

In multiplication, we add exponents. (Conversely, in division, we subtract exponents.) Therefore:

$$\begin{array}{l} 1.0 \times 10^2 \\ 2.0 \times 10^2 \\ \hline 2.0 \times 10^4 = 20{,}000 \end{array}$$

Floating-Point Numbers

Computers utilizing floating-point arithmetic were devised in order to avoid the problems involved in scaling. In floating-point computers, numbers are stored in a fixed format, including the number itself and the power of the base. There are some disadvantages in this type of computer, since additional equipment is required in order to perform floating-point arithmetic.

The use of floating-point numbers is, in general, quite similar to using a slide rule, where numbers are converted to powers of ten, since there is no radix point on the slide rule.

Thus, to multiply 451 and 372, first express the numbers as:

$$451 = 4.51 \times 10^2$$

$$372 = 3.72 \times 10^2$$

Read the product as 16.7, which is multiplied by 10^4, or:

$$16.7 \times 10^4$$

Thus, a number X is converted into $X = N \times 10^m$ as shown in Table 3-11.

Table 3-11. Conversion to Floating-Point Numbers

X	N	10^m
88.0	.88	10^2
4.6	.46	10^1
.14	.14	10^0
705.0	.705	10^3

Storage space for numbers within the computer can be utilized to the best advantage by moving the point to some finite number of positions to the left, let us say, 50 places. In this case, a number such as 828 would be represented as $.00\cdots0828 \times 10^{50}$. It follows that negative powers of ten will be indicated as exponents less than 50. In computers in use, which utilize floating-point arithmetic, the first two digits or the last two digits are reserved for the exponent. Also, within the computer the power of ten of a number is automatically noted, with the exponent being operated upon separately from the number. Correct shifting is made by the computer if, in adding and subtracting, the exponents are not the same. During multiplication the exponents are added; and, in division, they are subtracted.

SHIFT OPERATIONS

With a number such as 53746.942, a shift right to 5374.6942 is a division by 10; a shift left to 537469.42 is a multiplication by 10. Binary shifts correspond to division or multiplication by 2, as shown in Table 3-12.

Table 3-12. Binary Shifting

Shift	Binary Number	Decimal
	00001011010.0000	90
Left Once	00010110100.0000	180
Left Twice	00101101000.0000	360
Reset	00001011010.0000	90
Right Once	00000101101.0000	45
Right Twice	00000010110.1000	22.5

A computer shift register is used to multiply or divide, since shifting can be used for both operations. For example, consider 21_{10}. In the binary system, it is:

$$010101 = (0 \times 32 + 1 \times 16 + 0 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1) = 21_{10}$$

A shift left is:

$$101010 = (1 \times 32 + 0 \times 16 + 1 \times 8 + 0 \times 4 + 1 \times 2 + 0 \times 1) = 42_{10}$$

A decimal shift left is a multiply-by-ten operation, whereas a binary shift left, as above, is a multiply-by-two operation. In the same manner, the binary shift right is a divide-by-two operation:

$$010101.0 = 21_{10}$$

$$001010.1 = 1 \times 8 + 0 \times 4 + 1 \times 2 + 0 \times 1 + 1 \times .5 = 10.5_{10}$$

CHAPTER 4

Polynomials and Linear Forms

From the techniques of the various numbering systems in the first three chapters it is now possible to delve into some of the classical mathematical techniques and to see how these are handled on a modern digital computer. This chapter first discusses various polynomial forms and simple equations, simultaneous linear equations, and then the techniques for handling them follow. The binomial expansion is explained as a basic, well known technique, which has a number of applications to evaluation of simple expressions in a digital machine.

POLYNOMIAL FORMS

A polynomial in the variable x and of degree n is the function:

$$f(x) = a_0x^n + a_1x^{n-1} + a_2x^{n-2} + \cdots + a_{n-1}x + a_n$$

where

$a_0, a_1, \cdots, a_n$ are constants,
a_0 is not 0,
n is a positive integer.

Some examples of polynomials are:

$$3x^6 - 4x^5 + 17x^4 + 12x^3 - 7x^2 + 2x - 3$$
$$12x^4 + 6x^3 - x^2 - x + 15$$

The first is of degree 6 $(n = 6)$, whereas the second is of degree 4 $(n = 4)$. If we set $f(x) = 0$ we have an *integral rational equation* of degree n. Any root of $f(x) = 0$ is by definition zero of $f(x)$.

The roots of an equation (Fig. 4-1) are often necessary for the solution of a specific problem. For example, consider a polynomial, $f(x)$. If $f(x)$ is divided by $x - r$, for any $r \neq x$, the remainder R is equal to the value of the polynomial when x is replaced by r. If $Q(x)$ is the quotient, then:

$$\frac{f(x)}{x-r} = Q(x) + \frac{R}{x-r}$$

For $x = r$, this equation is:

$$\begin{aligned} f(x) &= (x-r)Q(x) + R \\ f(r) &= (r-r)Q(r) + R \\ f(r) &= R \end{aligned}$$

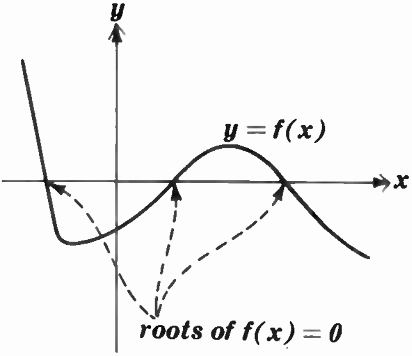


Fig. 4-1. Roots of the equation $y = f(x)$.

Consider $f(x) = x^3 + 2x^2 + x + 3$, and take $r = 2$. Then, dividing $f(x)$ by $x - 2$, we have:

$$\begin{array}{r|l}
 & x^2 + 4x + 9 \\ \hline
x - 2 & x^3 + 2x^2 + x + 3 \\
 & x^3 - 2x^2 \\ \cline{2-2}
 & \quad 4x^2 + x \\
 & \quad 4x^2 - 8x \\ \cline{2-2}
 & \qquad 9x + 3 \\
 & \qquad 9x - 18 \\ \cline{2-2}
 & \qquad\quad 21
\end{array}$$

Thus:

$$\begin{aligned} r &= 2 \\ R &= 21 \\ Q &= x^2 + 4x + 9 \end{aligned}$$

Substituting $x = r = 2$ in $f(x)$ gives:

$$\begin{gathered} 2^3 + 2(2)^2 + 2 + 3 = 21 \\ f(r) = 21 \\ R = 21 \end{gathered}$$

Factors—A factor is one of several values whose product equals the number in question.

Thus, the factors of 12 are:

$$\begin{aligned} 12 \times 1 &= 12 \\ 6 \times 2 &= 12 \\ 3 \times 4 &= 12 \\ 3 \times 2 \times 2 &= 12 \end{aligned}$$

Note that if r is a root of $f(x) = 0$, then $f(r) = 0$, and therefore $R = 0$. Thus:

$$f(x) = (x - r)\, Q(x)$$

That is, $x - r$ is a factor of $f(x)$.

Conversely, if $x - r$ is a factor of $f(x)$, then:

$$f(x) = (x - r)\, P(x)$$

where $P(x)$ is the other factor(s) of $f(x)$.

Then:

$$\frac{f(x)}{x - r} = P(x)$$

And thus $R = 0$. But $f(r) = R = 0$. Hence r is a root of $f(x) = 0$. We have just proved that r is a root of $f(x) = 0$ if, and only if, $x - r$ is a factor of $f(x)$.

Often division by $x - r$ is required in evaluating the roots of $f(x)$; hence a short-cut technique of division is useful. For example, consider the division of $2x^3 - 3x^2 - 13x + 5$ for $r = 3$. In the long-division manner, this is:

$$\begin{array}{r|rrrr}
 & 2x^2 & +\ 3x & -\ 4 & \\ \cline{2-5}
x - 3 & 2x^3 & -\ 3x^2 & -\ 13x & +\ 5 \\
 & 2x^3 & -\ 6x^2 & & \\ \cline{2-3}
 & & 3x^2 & -\ 13x & \\
 & & 3x^2 & -\ 9x & \\ \cline{3-4}
 & & & -\ 4x & +\ 5 \\
 & & & -\ 4x & +\ 12 \\ \cline{4-5}
 & & & & -\ 7
\end{array}$$

However, a shorter method, known as *synthetic division*, may be used. In general, if $f(x) = a_0 x^n + \cdots + a_{n-1} x + a_n$ is a polynomial of degree n, then $f(x)$ divided by $x - r$, for any real number r, is given by:

$$\frac{f(x)}{x-r} = A_0 x^{n-1} + A_1 x^{n-2} + \cdots + A_{n-1} + \frac{R}{x-r}$$

where, for $i = 1, \cdots, n$,

$$A_0 = a_0,$$
$$A_i = A_{i-1}\, r + a_i$$
$$A_n = R \text{ (remainder).}$$

This is commonly written as:

$$\begin{array}{r|llll} 1 - r & a_0 & a_1 & a_2 \cdots & a_n \\ & & a_0 r & A_1 r \cdots & A_{n-1} r \\ \hline & a_0 & A_1 & A_2 \cdots & R \end{array}$$

In the preceding example for $r = 3$, $f(x) = 2x^3 - 3x^2 - 13x + 5$, this is:

$$\begin{array}{r|rrrr} 1 - 3 & 2 & -3 & -13 & +5 \\ & & -6 & -9 & +12 \\ \hline & 2 & +3 & -4 & -7 \end{array}$$

The expression $2 + 3 - 4 - 7$ is understood to mean $2x^2 + 3x - 4 = Q(x)$, while $R = -7$. Any missing term is written as with a zero coefficient, for each power of x is in descending order. For example, $6x^4 + 7x^3 + 3x + 17$ is written as $6x^4 + 7x^3 + 0x^2 + 3x + 17$.

Here are some rules for determining the roots of an equation.

1. Every integral rational equation of degree n has n roots (but the same root may occur more than once).
2. If an integral rational equation $f(x) = 0$, with real coefficients, has one root of the form $a + bi$ ($i^2 = -1$), it also has the conjugate $(a - bi)$ as a root.
3. An integral rational equation $f(x) = 0$, with real coefficients, has the same number of positive roots as it has variations of signs (or less by an even number), where some roots may be multiple.

For example, consider:

$$3x^4 + 7x^3 - 8x^2 + 9x + 7 = 0 \qquad (1)$$

$$7x^5 - 3x^4 + 2x^3 + x^2 - 12x + 14 = 0 \qquad (2)$$

By rule 1, equation 1 has four roots, and by rule 3 it has either two positive roots or no positive roots. Equation 2 has five roots, of which four, two, or none are positive roots.

The equation in Fig. 4-2 has two real roots. In Fig. 4-3, $y = f(x)$ has three real roots; y' has two real roots, and y'' has one real root.

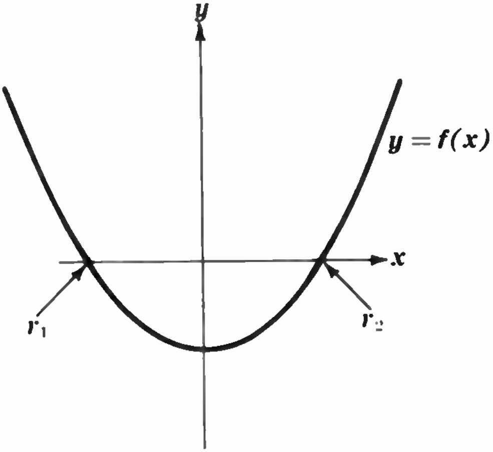


Fig. 4-2. The real roots (r_1, r_2) of a parabola $y = f(x)$.

Evaluation of Polynomials

Since synthetic division may be performed with fewer steps than most methods, it is a type of factoring often used to evaluate a polynomial. These steps generally include successive multiplications and additions.

Any analytic function $f(x)$ may be written as a Taylor's series. For example, consider:

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots$$

Thus, if sin 0.5 is required, the series is:

$$\sin 0.5 = 0.5 - \frac{0.5^3}{3!} + \frac{0.5^5}{5!} - \frac{0.5^7}{7!} + \cdots$$

$$= 0.5 - 0.0208333 + 0.00026042 - 0.00000155 + \cdots$$

$$\sin 0.5 = 0.4794$$

This is accurate enough for a specific case, and if for example we wanted sin x to within 0.0001, the series is used until a term less than 0.0001 is reached.

However, synthetic division provides a simplification. The series for e^x is as follows:

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \frac{x^6}{6!} + \frac{x^7}{7!} + \frac{x^8}{8!} + \cdots$$

1. Multiply the coefficient of x^N, the highest power of x, by x. This gives:

$$\frac{1}{8!}x$$

2. Next, the coefficient of the next highest power of x is added to produce:

$$\frac{1}{8!}x + \frac{1}{7!}$$

And this is again multiplied by x to obtain:

$$\left(\frac{1}{8!}x + \frac{1}{7!} \right)x = \frac{x}{7!} + \frac{x^2}{8!}$$

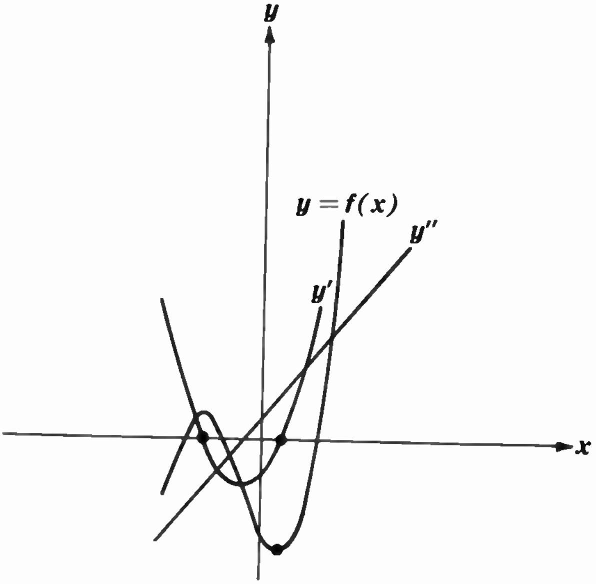


Fig. 4-3. Graph of $y = x^3 + 3x^2 - 10x - 24$.

3. The next lowest coefficient is added, and the expression is again multiplied by x:

$$\left(\frac{1}{6!} + \frac{x}{7!} + \frac{x^2}{8!} \right)x = \frac{x}{6!} + \frac{x^2}{7!} + \frac{x^3}{8!}$$

4. The succeeding step results in:

$$\left(\frac{1}{5!} + \frac{x}{6!} + \frac{x^2}{7!} + \frac{x^3}{8!} \right)x = \frac{x}{5!} + \frac{x^2}{6!} + \frac{x^3}{7!} + \frac{x^4}{8!}$$

5. When the highest desired power of x has been reached (which is x^8 in this case), the polynomial is of the form:

$$x+\frac{x^2}{2!}+\frac{x^3}{3!}+\frac{x^4}{4!}+\frac{x^5}{5!}+\frac{x^6}{6!}+\frac{x^7}{7!}+\frac{x^8}{8!}$$

And when 1 is added to this, the value for e has been obtained.

This is faster and requires fewer steps than the more common way of evaluating the approximation of e^x that is given below:

1. To 1 add x.
2. Multiply x by x and divide by 2!
3. Add (1) and (2) above to obtain the sum of the first three terms.
4. Take (2) above, multiply it by x and divide by 3.
5. Add (4) to (3). This result is the sum of the first four terms.
6. This continues for all required steps.

The same general approach can be used for the cos x series:

$$\cos x=1-\frac{x^2}{2!}+\frac{x^4}{4!}-\frac{x^6}{6!}+\frac{x^8}{8!}+\cdots$$

$$\left(\frac{1}{8!}\right)x^2=\frac{x^2}{8!}$$

$$\left(\frac{x^2}{8!}-\frac{1}{6!}\right)x^2=\frac{x^4}{8!}-\frac{x^2}{6!}$$

$$\left(\frac{x^4}{8!}-\frac{x^2}{6!}+\frac{1}{4!}\right)x^2=\frac{x^6}{8!}-\frac{x^4}{6!}+\frac{x^2}{4!}$$

$$\left(\frac{x^6}{8!}-\frac{x^4}{6!}+\frac{x^2}{4!}-\frac{1}{2!}\right)x^2=\frac{x^8}{8!}-\frac{x^6}{6!}+\frac{x^4}{4!}-\frac{x^2}{2!}$$

Adding 1 and rearranging the terms gives the desired form:

$$\cos x=1-\frac{x^2}{2!}+\frac{x^4}{4!}-\frac{x^6}{6!}+\frac{x^8}{8!}$$

SUBROUTINES FOR EVALUATION

The evaluations, as before, can be used as programming loops or subroutines in digital computers. Rather than storing long tables in a limited computer memory, this technique is often used. Two examples are given: one for square root, and one for $\log_e x$.

Mathematical Description of Square-Root Subroutine—We wish to find $\sqrt{N}$ for $0 < N < 1$. First, normalize N so that $N' = 2^n N$ *and* $\frac{1}{2} \leqq N' < 1$, for some integer n. Now, if n is even, we have:

$$\sqrt{N} = 2^{-n/2}\sqrt{N'}$$

If n is odd, then we have:

$$\sqrt{N} = 2^{-\frac{(n-1)}{2}}\sqrt{\frac{N'}{2}}$$

In the following discussion, let N^* denote N' if n is even, or $\frac{N'}{2}$ if n is odd. The first estimate of $\sqrt{N^*}$ is E_0, which is given by:

$$E_0 = \frac{1}{2} + \frac{N^*}{4}$$

The next estimates are:

$$E_1 = \frac{N^*}{2E_0} + \frac{E_0}{2}$$

$$E_2 = \frac{N^*}{2E_1} + \frac{E_1}{2}$$

$$\cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot$$

$$E_k = \frac{N^*}{2E_{k-1}} + \frac{E_{k-1}}{2}$$

It follows that:

$$\sqrt{N^*} \approx E_k \text{ if } | E_k - E_{k-1} | \leqq 2^{-35}$$

Natural Logarithm—Evaluation of the logarithm function may be made by using a simple expansion. Hence, for $\log_e x$, we have:

$$\log x = 2\left[\frac{x-1}{x+1} + \frac{1}{3}\left(\frac{x-1}{x+1}\right)^3 + \frac{1}{5}\left(\frac{x-1}{x+1}\right)^5 + \cdots\right] \qquad x > 0$$

$$\log x = \frac{x-1}{x} + \frac{1}{2}\left(\frac{x-1}{x}\right)^2 + \frac{1}{3}\left(\frac{x-1}{x}\right)^3 + \cdots \qquad x > \frac{1}{2}$$

$$\log x = (x-1) - \frac{(x-1)^2}{2} + \frac{(x-1)^3}{3} - \cdots \qquad 0 < x \leqq \frac{1}{2}$$

HIGHER-DEGREE EQUATIONS

The second-degree equation (or quadratic), which is a parabola, has a pair of roots. Consider:

$$x = \frac{-b \pm \sqrt{b^2 - 4\,ac}}{2a}$$

This is the solution of the general equation $ax^2 + bx + c = 0$. In this solution, $b^2 - 4ac$ is termed the *discriminant (D)* of the equation. The roots of the quadratic equation may be characterized by the sign of D: If D is greater than 0, the roots are real and unequal; if D is equal to 0, the roots are real and equal; if D is less than 0, the roots are conjugate imaginaries.

Consider the equation $4x^2 - 3x + 2 = 0$. In this case:

$$D = 9 - 4(8)$$
$$D = -23$$

By the preceding rules the roots are of the form $a \pm bi$. Therefore, the roots are:

$$x = \frac{3 \pm (-23)^{1/2}}{8}$$

or,

$$x = \frac{3 + i(23)^{1/2}}{8}, \frac{3 - i(23)^{1/2}}{8}$$

The Cubic Equation

The general form of a cubic equation is $y^3 + py^2 + qy + r = 0$. This equation may be reduced to:

$$x^3 + ax + b = 0$$

This is done by substituting for y the relation:

$$y = x - \frac{p}{3}$$

Therefore:

$$(x - \frac{p}{3})^3 + p(x - \frac{p}{3})^2 + q(x - \frac{p}{3}) + r = 0$$

From this, we derive:

$$a = \frac{3q - p^2}{3}$$

$$b = \frac{2p^3 - 9pq + 27r}{27}$$

We define A and B as:

$$A \equiv \sqrt[3]{-\frac{b}{2}+\sqrt{\frac{b^2}{4}+\frac{a^3}{27}}}$$

$$B \equiv \sqrt[3]{-\frac{b}{2}-\sqrt{\frac{b^2}{4}+\frac{a^3}{27}}}$$

From this there are three values for x:

$$x_1 = A + B$$

$$x_2 = -\frac{A+B}{2}+\frac{A-B}{2}\sqrt{-3}$$

$$x_3 = -\frac{A+B}{2}-\frac{A-B}{2}\sqrt{-3}$$

We may define the *discriminant (D)* of a cubic equation by:

$$D = \frac{b^2}{4}+\frac{a^3}{27}$$

If D is greater than 0, one root is real and two are complex conjugates. If D is equal to 0, all three roots are real, and at least two are equal. If D is less than 0, all three roots are real and unequal.

For another approach, consider the following:

$$f(x) = x^3 - 3x^2 - 12x + 1 = 0$$

Table 4-1 gives values for $f(x)$ determined by various values of x.

Table 4-1. Values of $f(x)$

x	−4	−2	−1	0	1	2	3	8
$f(x)$	−63	5	9	1	−15	−27	−35	225

There are three real roots. This may be seen from Table 4-1. Note that $f(x)$ must equal zero for some value of x between $x = -4$ and $x = -2$; for some value of x between $x = 0$ and $x = 1$; and for some value of x between $x = 3$ and $x = 8$. Each of these roots may be approximated. Consider the root between $x = 0$ and $x = 1$; let us call the first approximation d (Table 4-2).

By the straight-line formula of analytic geometry, we have:

$$d = \frac{x_1 - x_2}{y_2 - y_1} = \frac{0-1}{-15-1} = \frac{1}{16}$$

$$= 0.06$$

Table 4-2. Approximation of x

x	1	0+d	1
f(x)	1	0	−15

By synthetic division:

$$\begin{array}{r|rrrr} .06 & 1 & -3.00 & -12.00 & +1.00 \\ & & +0.06 & +\ 1.76 & -1.76 \\ \hline & 1 & +2.49 & -10.24 & -0.76 \end{array}$$

and,

$$\begin{array}{r|rrrr} .07 & 1 & -3.00 & -12.00 & +1.00 \\ & & +0.07 & -\ 2.05 & -0.98 \\ \hline & 1 & -2.93 & -14.05 & +0.02 \end{array}$$

Thus this desired root is between 0.06 and 0.07. Table 4-3 lists values of $f(x)$ for making the next approximation.

$$\frac{d^{(1)}}{0.07-0.06}=\frac{0-(-0.76)}{0.02-(-0.76)}$$

$$\frac{d^{(1)}}{0.01}+\frac{0.76}{0.78}$$

$$d^{(1)}=0.0097$$

Since the earlier approximation was 0.06, the next approximation is 0.06 + 0.0097, or 0.0697.

Table 4-3. The Second Approximation of x for which f(x) = 0

x	−0.76	0	0.02
f(x)	0.06	0.06 + $d^{(1)}$	0.07

SIMULTANEOUS LINEAR EQUATIONS

Many examples can be given of a fixed series of steps that will solve a class of problems. An example is simultaneous linear equations. Consider the simple case of two equations:

$$2x + 4y = -10 \qquad (1)$$

$$3x - 9y = \quad 45 \qquad (2)$$

These could be solved by equating each for x:

$$x = -\ 5 - 2y \qquad (3)$$

$$x = \quad 15 + 3y \qquad (4)$$

And equating these to solve for y:

$$-5 - 2y = 15 + 3y$$
$$-5y = 20$$
$$y = -4$$

Substituting $y = -4$ in (3) gives the solution:

$$x = -5 + 8$$

The complete solution is:

$$x = 3$$
$$y = -4$$

Fig. 4-4 shows the graphical solution of this intersection.

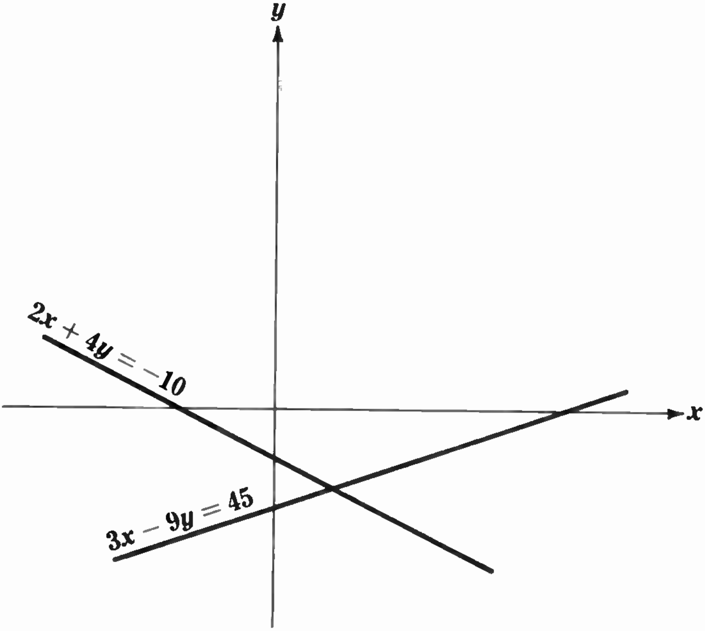


Fig. 4-4. Graph solution of text problem.

A second method involves removing the x term by multiplying (1) by 3 and (2) by 2:

$$6x + 12y = -30 \qquad (5)$$
$$6x - 18y = 90 \qquad (6)$$

Subtracting (6) from (5) yields:

$$30y = -120$$
$$y = -4$$

By substitution of $y = -4$ in (5), we obtain:

$$6x + 12(-4) = -30$$
$$6x = \quad 18$$

Hence:

$$x = \quad 3$$
$$y = \quad -4$$

Neither method can be used for a series of n linear equations in n unknowns (where n is greater than 2) in any direct and easily mechanized system that can be applied in the general case. But the second method can be expanded into a usable technique. Consider the following system of equations:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ \cdots\cdots\cdots\cdots&\cdots\cdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n \end{aligned}$$

These can be reduced by one unknown at a time to obtain a complete solution to the system. For example, consider:

$$2w - 2x + 6y - 2z = \quad 8 \qquad (1)$$
$$3w + 3x + 4y - 6z = -3 \qquad (2)$$
$$7w - \quad x + 4y + 2z = \ 25 \qquad (3)$$
$$8w + 4x + 2y - 3z = \ 10 \qquad (4)$$

Divide both sides of (1) by 2 so that:

$$w - \quad x + 3y - \quad z = \quad 4 \qquad (5)$$

We wish to eliminate the w-term, so we multiply (5) by 3 and subtract this from (2):

$$\begin{array}{r} 3w + 3x + 4y - 6z = - \ 3 \\ \underline{3w - 3x + 9y - 3z = \quad 12} \\ 6x - 5y - 3z = -15 \end{array} \qquad (6)$$

In the same manner we multiply (5) by 7 and subtract this from (3):

$$\begin{array}{r} 7w - \quad x + \quad 4y + 2z = \ 25 \\ \underline{7w - 7x + 21y - 7z = \ 28} \\ 6x - 17y + 9z = -3 \end{array} \qquad (7)$$

Also in the same manner we multiply (5) by 8 and subtract this from (4):

$$\begin{array}{r} 8w + 4x + 2y - 3z = 10 \\ 8w - 8x + 24y - 8z = 32 \\ \hline 12x - 22y + 5z = -22 \end{array} \qquad (8)$$

This results in three equations in three unknowns:

$$6x - 5y - 3z = -15 \qquad (6)$$

$$6x - 17y + 9z = -3 \qquad (7)$$

$$12x - 22y + 5z = -22 \qquad (8)$$

Now, subtracting (6) from (7), we have:

$$-12y + 12z = 12 \qquad (9)$$

And multiplying (6) by 2 and then subtracting this result from (8):

$$\begin{array}{r} 12x - 22y + 5z = -22 \\ 12x - 10y - 6z = -30 \\ \hline -12y + 11z = 8 \end{array} \qquad (10)$$

Now there are two equations in two unknowns:

$$-12y + 12z = 12 \qquad (9)$$

$$-12y + 11z = 8 \qquad (10)$$

Subtracting (10) from (9) gives:

$$z = 4 \qquad (11)$$

To evaluate these equations for w, x, and y, substitute (11) in (10):

$$-12y = -36$$

Or.

$$y = 3$$

From (8), substitution for y and z yields:

$$12x - 66 + 20 = -22$$

or,

$$x = 2$$

In the same manner, by using (5) we have:

$$w - 2 + 9 - 4 = 4$$

or,

$$w = 1$$

The set of equations which are actually solved is:

$$w - x + 3y - z = 4 \tag{5}$$
$$6x - 5y - 3z = -15 \tag{6}$$
$$-12y + 12z = 12 \tag{9}$$
$$z = 4 \tag{11}$$

The complete solution is:

$$x = 2 \qquad w = 1$$
$$y = 3 \qquad z = 4$$

A general solution of this type can easily be automated or programmed for use by a computer.

DETERMINANTS

The example previously given for the solution of n simultaneous equations in n unknowns implicitly uses determinants.

Consider two equations in two unknowns:

$$a_{11}x_1 + a_{12}x_2 = b_1 \tag{1}$$
$$a_{21}x_1 + a_{22}x_2 = b_2 \tag{2}$$

Multiplying (1) by a_{22}, and (2) by a_{12}:

$$a_{22}a_{11}x_1 + a_{22}a_{12}x_2 = a_{12}b_1 \tag{3}$$
$$a_{21}a_{12}x_1 + a_{22}a_{12}x_2 = a_{12}b_2 \tag{4}$$

Thus:

$$x_1 = \frac{a_{22}b_1 - a_{12}b_2}{a_{22}a_{11} - a_{21}a_{12}}$$
$$x_2 = \frac{a_{11}b_2 - a_{21}b_1}{a_{22}a_{11} - a_{21}a_{12}}$$

The terms in the denominators (which are the same) may be expressed in an array as:

$$\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}$$

This is defined as being equal to $a_{11}a_{22} - a_{12}a_{21}$.

Thus, following this rule, we can write:

$$\begin{vmatrix} 3 & 4 \\ -7 & 2 \end{vmatrix} = 6 - (-28) = 34$$

And, in the same manner, we have:

$$\begin{vmatrix} 2 & 5 \\ 4 & 6 \end{vmatrix} = 12 - 20 = -8$$

The diagonal from upper left to lower right is called the *principal diagonal*; the determinant's value is the product of the principal diagonal minus the product of the other diagonal. The use of determinants permits a rapid solution of simultaneous equations.

Consider the general equations (1) and (2). The solutions are:

$$x_1 = \frac{\begin{vmatrix} b_1 & a_{12} \\ b_2 & a_{22} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}}, \qquad \text{and} \qquad x_2 = \frac{\begin{vmatrix} a_{11} & b_1 \\ b_{21} & b_2 \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}}$$

For example, consider the following equations:

$$\begin{aligned} 2x - 3y &= 16 \\ 5x + 2y &= \ \ 2 \end{aligned}$$

Then:

$$x = \frac{\begin{vmatrix} 16 & -3 \\ 2 & 2 \end{vmatrix}}{\begin{vmatrix} 2 & -3 \\ 5 & 2 \end{vmatrix}} = \frac{32 - (-6)}{4 - (-15)} = \frac{38}{19}$$

$$x = 2$$

$$y = \frac{\begin{vmatrix} 2 & 16 \\ 5 & 2 \end{vmatrix}}{\begin{vmatrix} 2 & -3 \\ 5 & 2 \end{vmatrix}} = \frac{4 - 80}{4 - (-15)} = \frac{-76}{19}$$

$$y = -4$$

One can operate on the determinant with specific rules to simplify their solution.

(1) The determinant's value is unchanged if corresponding rows and columns are interchanged.

From the previous example, by interchanging the first row and first column of the determinant in the numerator and interchanging the first row and first column in the denominator, we have:

$$x = \frac{\begin{vmatrix} 16 & 2 \\ -3 & 2 \end{vmatrix}}{\begin{vmatrix} 2 & 5 \\ -3 & 2 \end{vmatrix}} = \frac{32 - (-6)}{4 - (-15)} = \frac{38}{19}$$

$$x = 2$$

(2) If two columns or rows are interchanged, the sign of the determinant is changed.

Interchanging the first and second columns of the determinant in the numerator of x gives:

$$x = \frac{\begin{vmatrix} -3 & 16 \\ 2 & 2 \end{vmatrix}}{\begin{vmatrix} 2 & -3 \\ 5 & 2 \end{vmatrix}} = \frac{-6 - 32}{4 - (-15)}$$

$$x = -2$$

(3) If all the numbers in a row or column are multiplied by a constant k, the value of the determinant is multiplied by this constant.

Multiplying by 3 the second column of the determinant in the numerator of x:

$$3x = \frac{\begin{vmatrix} 16 & -9 \\ 2 & 6 \end{vmatrix}}{\begin{vmatrix} 2 & -3 \\ 5 & 2 \end{vmatrix}} = \frac{96 - (-18)}{4 - (-15)}$$

$$3x = \frac{114}{19} = 6$$

(4) If the numbers in a row or column are added to another row or column, the determinant is unchanged in value:

$$x = \frac{\begin{vmatrix} 16 & -3 \\ 2 & 2 \end{vmatrix}}{\begin{vmatrix} 2 & -3 \\ 5 & 2 \end{vmatrix}}$$

By Rule 4:

$$x = \frac{\begin{vmatrix} 16 & 13 \\ 2 & 4 \end{vmatrix}}{\begin{vmatrix} 2 & -1 \\ 5 & 7 \end{vmatrix}}$$

Hence:

$$x = \frac{64 - 26}{14 - (-5)} = \frac{38}{19}$$

$$x = 2$$

Using these and other rules, one can reduce a determinant of order greater than 2 in order to allow more simple evaluation.

Determinants of Higher Order

As before, the second-order determinant is expanded as:

$$\begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix} = a_1b_2 - a_2b_1$$

For the third-order determinant, expansion is possible either directly or by means of minors. By direct expansion, we have:

$$\begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix} = a_1b_2c_3 + a_2b_3c_1 + a_3b_1c_2 - a_3b_2c_1 - a_2b_1c_3 - a_1b_3c_2$$

Minors can also be used for expansion; consider only the terms having a_1 as a coefficient. These are $a_1b_2c_3 - a_1b_3c_2$. This difference is defined as equal to:

$$a_1 \begin{vmatrix} b_2 & c_2 \\ b_3 & c_3 \end{vmatrix}$$

The expansion by minors is:

$$\begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix} = a_1 \begin{vmatrix} b_2 & c_2 \\ b_3 & c_3 \end{vmatrix} - a_2 \begin{vmatrix} b_1 & c_1 \\ b_3 & c_3 \end{vmatrix} + a_3 \begin{vmatrix} b_1 & c_1 \\ b_2 & c_2 \end{vmatrix}$$

This is also equal to:

$$-b_1 \begin{vmatrix} a_2 & c_2 \\ a_3 & c_3 \end{vmatrix} + b_2 \begin{vmatrix} a_1 & c_1 \\ a_3 & c_3 \end{vmatrix} - b_3 \begin{vmatrix} a_1 & c_1 \\ a_2 & c_2 \end{vmatrix}$$

This is, in turn, also equal to:

$$+c_1 \begin{vmatrix} a_2 & b_2 \\ a_3 & b_3 \end{vmatrix} - c_2 \begin{vmatrix} a_3 & b_3 \\ a_1 & b_1 \end{vmatrix} + c_3 \begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix}$$

THE BINOMIAL EXPANSION

In general, it is possible to expand $(a+b)^n$ as:

$$(a+b)^n = a^n + \frac{n}{1} a^{n-1} b + \frac{n(n-1)}{2!} a^{n-2} b^2$$
$$+ \frac{n(n-1)(n-2)}{3!} a^{n-3} b^3 + \cdots + b^n$$

Thus, for $(a+3)^4$ one can write:

$$(a+3)^4 = a^4 + 4a^3 (3) + \frac{4(3)}{2} a^2 (3)^2 + \frac{4(3)(2)}{3\cdot 2} a (3)^3 + 3^4$$
$$= a^4 + 12a^3 + 54a^2 + 108a + 81$$

The binomial expansion can be used for calculation of certain values, such as the following:

$$\begin{aligned} (102)^4 &= (100+2)^4 \\ &= 100^4 + 4(100)^3 (2) + \frac{4(3)}{2} 100^2 (2)^2 \\ &\quad + \frac{4(3)(2)}{2\cdot 3} 100 (2)^3 + 2^4 \\ &= \begin{array}{r} 100{,}000{,}000 \\ 8{,}000{,}000 \\ 240{,}000 \\ 3{,}200 \\ 16 \\ \hline \end{array} \\ (102)^4 &= 108{,}243{,}216 \end{aligned}$$

In the same manner, one can evaluate, say, $(506)^6 = (500+6)^6$, or $(1.4)^4 = (1.0 + 0.4)^4$.

Consider the binomial expansion term with b^r; this is the $r+1$st term in the expansion. In general, this term can be expressed as:

$$\frac{n(n-1)(n-2)\cdots(n-r+1)}{r!} a^{n-r} b^r$$

As an example, find the third term of $(2x - 3y)^{12}$. The third term will have $r = 2$:

$$\frac{12(12-2+1)}{2!}(2x)^{12-2}(-3y)^2 = 607{,}256x^{10}y^2$$

Coefficients for the binomial expansion can be expressed in *Pascal's Triangle* (Table 4-4) where the first row represents the coefficients of $(a+b)^0$; the second row represents the coefficients of $(a+b)^1$; the third, $(a+b)^2$, and so on. Any number can be determined by adding the two adjacent numbers above it, as $1 + 6 = 7$, $15 + 20 = 35$, $21 + 35 = 56$, and so on. If this is applied for $a = 1$ and $b = x$, the binomial series becomes:

$$(1+x)^n = 1 + nx + \frac{n(n-1)}{2!}x^2 + \frac{n(n-1)(n-2)}{3!}x^3 + \cdots + x^n$$

For example, consider $(1+x)^3$:

$$(1+x)^3 = 1 + 3x + \frac{3\ (2)}{2}x^2 + \frac{3(2)\ (1)}{6}x^3$$
$$= 1 + 3x + 3x^2 + x^3$$

If $n = -2$, for example, we have the infinite series:

$$(1+x)^{-2} = 1 - 2x + \frac{-2(-3)}{2}x^2 + \frac{-2(-3)(-4)}{3 \cdot 2}x^3 + \cdots$$
$$= 1 - 2x + 3x^2 - 4x^3 + \cdots$$

Consider $\sqrt{21} = (25-4)^{1/2}$. To place this in the $(1+x)^n$ form, we express it as:

$$\sqrt{21} = [\ 25\ (1 - \frac{4}{25})\ ]^{1/2}$$
$$= 5\ (1 - \frac{4}{25})^{1/2}$$
$$= 5\ [\ 1 + 1/2(-.16) + \frac{1/2(-1/2)}{2}(.16)^2\ ]$$
$$= 5\ [\ 1 - .08 - .0032\ ]$$
$$= 5\ [\ 1 - .0832\ ]$$
$$= 5\ [\ .9168\ ] = 4.5840$$

Table 4-4. Pascal's Triangle

								1								
							1		1							
						1		2		1						
					1		3		3		1					
				1		4		6		4		1				
			1		5		10		10		5		1			
		1		6		15		20		15		6		1		
	1		7		21		35		35		21		7		1	
1		8		28		56		70		56		28		8		1

In the same manner, one can find the root of 1.2:

$$\begin{aligned}\sqrt{1.2} &= (1 + 0.2)^{1/2}\\ &= 1 + 1/2\ (0.2) + \frac{1/2\ (-1/2)}{2}\ (0.2)^2 + \cdots\\ &= 1 + 0.1 - 0.005 + \cdots\\ &= 1.095\end{aligned}$$

Note that with one term, the root is 1; to two terms, it is 1.1; to three terms, it is 1.095. Thus, the value approaches the correct value as more terms are used.

CHAPTER 5

Interpolation

In a general sense, interpolation involves finding a certain value of a function by using the values that this function assumes for specific values of the independent variable. The mathematical relation between the independent and dependent variables is itself not known. This, then, is actually a method of curve-fitting, and for most curves a good fit can be found.

INTRODUCTION

In 1885, the German mathematician Karl Theodor Weierstrass proved the following two theorems:

1. Any function $f(x)$ that is continuous over the interval (a,b) can always be represented (approximated) over this interval by some polynomial. That is, there exists a polynomial $P(x)$ such that $|f(x) - P(x)|$ is less than ϵ for every value of x in the interval (a,b) and where ϵ can be *any* preassigned positive number (Fig. 5-1).
2. Any periodic continuous function $f(x)$ with period 2π can be represented by a trigonometric series whose form is:

$$\begin{aligned} g(x) &= a_0 + a_1 \sin x + a_2 \sin 2x + a_3 \sin 3x + \cdots \\ &\quad + b_1 \cos x + b_2 \cos 2x + b_3 \cos 3x + \cdots \\ &= a_0 + \sum_{n=0}^{\infty} (a_n \sin nx + b_n \cos nx) \end{aligned}$$

where,

$|f(x) - g(x)| < \epsilon$,

ϵ is any preassigned positive number.

The first theorem means that one can always find or determine a function to fit data which are continuous; the second means that for approximations a trigonometric series is often very valuable. For this reason some basic trigonometric functions will be reviewed.

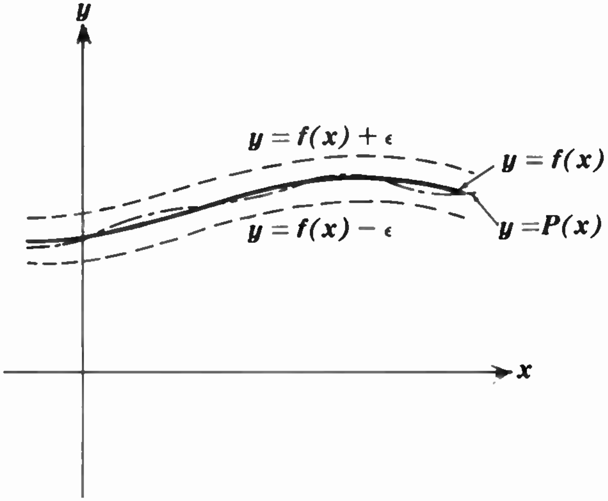


Fig. 5-1. Illustration of Weierstrass' Theorem.

TRIGONOMETRIC FUNCTIONS

Basic trigonometric functions (sine, cosine, and tangent) can be stored in the computer memory, although this is wasteful of valuable memory space. External (magnetic tape) memories may be used to store trigonometric tables, but there are other techniques.

Consider the relations between the trigonometric functions of an angle:

$$\sin^2\theta + \cos^2\theta = 1 \qquad (1)$$

$$1 + \tan^2\theta = \sec^2\theta \qquad (2)$$

$$1 + \cot^2\theta = \csc^2\theta \qquad (3)$$

These relations are well known and can be used to find one function if the other is given as sine or cosine in (1), tangent or secant (2), or cotangent or cosecant (3).

As in the preceding chapter, the series expression for sin x can be evaluated as:

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots$$

One can also develop all necessary functions, given a single function. For example, given sin θ, we can write:

$$\cos\theta = \sqrt{1-\sin^2\theta} = \frac{1}{\sec\theta}$$

$$\tan\theta = \frac{\sin\theta}{\sqrt{1-\sin^2\theta}} = \frac{1}{\cot\theta}$$

$$\csc\theta = \frac{1}{\sin\theta}$$

Solutions to trigonometric equations require finding an angle to satisfy the given equation. Two simple equations will serve as examples:

$$\sin^2\theta - \frac{3}{2}\sin\theta + \frac{1}{2} = 0$$

$$(\sin\theta - \frac{1}{2})\ (\sin\theta - 1) = 0$$

$$\sin\theta = \frac{1}{2}, \qquad \text{and} \qquad \sin\theta = 1$$

$$\sin^{-1}\frac{1}{2} = 30°, \qquad \text{and} \qquad \sin^{-1}1 = 90°$$

Consider another example:

$$\cos^2\theta + 2\cos\theta - 3 = 0$$

$$(\cos\theta + 3)\ (\cos\theta - 1) = 0$$

$$\cos\ \theta = 1, \qquad \text{and} \qquad \cos\theta = -3$$

But $\cos\theta = -3$ is an impossible value; hence, the only solution is $\cos\theta = 1$.

Reduction of Angles—Instead of the full set of values of trigonometric functions, a restricted range of values suffices for many applications. By reducing the given angle to one less than 90°, many angles can be used, given only a small tabular list of functions. This reduction is made as follows:

1. Express the given angle β as 90° ± θ, where θ is less than 90°, and n is some positive integer (Fig. 5-2).
2. If n is even, write the given function of β as a function of θ. If n is odd, write the *cofunction* of the given function as a function of θ.
3. The sign of the angle is determined by (a) the function of the angle and (b) the quadrant in which the angle is located.

4. If the original angle β is negative, then the expression for any trigonometric function may be derived from $\sin(-\theta) = -\sin\theta$, and $\cos(-\theta) = \cos\theta$.

Consider the following examples:

$$\begin{aligned}\cos 300^\circ &= \cos(3 \times 90 + 30)^\circ \\ &= \sin 30^\circ \\ \sin 370^\circ &= \sin(4 \times 90 + 10)^\circ \\ &= \sin 10^\circ\end{aligned}$$

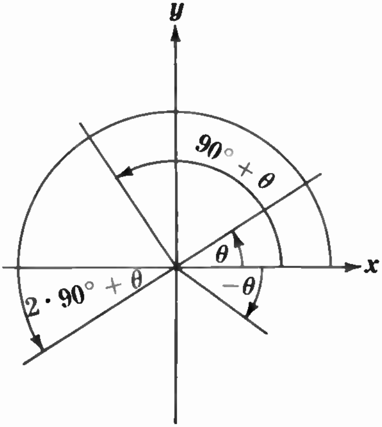


Fig. 5-2. Representation of $n90^\circ \pm \theta$.

Another way of expressing the reduction of angles is $\sin x = \cos(90 - x) = \sin(180 - x)$. For example, we have:

$$\sin 30^\circ = \cos 60^\circ = \sin 150^\circ$$
$$\sin 20^\circ = \cos 70^\circ = \sin 160^\circ$$

And:

$$\cos x = \sin(90 - x) = -\cos(180 - x)$$
$$\tan x = \cot(90 - x) = -\tan(180 - x)$$

Often it is useful to express the function of an angle as a sum of two angles. For example:

$$\sin(x + y) = \sin x \cos y + \cos x \sin y$$
$$\cos(x + y) = \cos x \cos y - \sin x \sin y$$
$$\tan(x + y) = \frac{\tan x + \tan y}{1 - \tan x \tan y}$$

For the differences between two angles, rather than their sum, the expressions on the right have all signs (shown) changed.

Further, functions of multiple angles can be expressed in terms of a single angle if that particular function of a single angle is known.

$$\begin{aligned}\sin 2x &= 2\sin x\cos x\\ \cos 2x &= 2\cos^2 x - 1\\ \sin 3x &= 3\sin x - 4\sin^3 x\\ \cos 3x &= 4\cos^3 x - 3\cos x\end{aligned}$$

DIFFERENCES

Interpolation of intermediate values in an equation is a powerful tool. The fundamentals of interpolation require consideration of the table of differences of the function.

Consider a function $f(x)$ whose values for equidistant points $x, x + h, x + 2h, \cdots, x + mh$ of the independent variable x are known. If we subtract $f(x)$ from $f(x+h)$, we establish the *delta* of f of x, of $\Delta f(x)$, where Δ is the symbol for the difference. In this way:

$$\begin{aligned}\Delta f(x) &= f(x+h) - f(x)\\ \Delta f(x+h) &= f(x+2h) - f(x+h)\\ \Delta f(x+2h) &= f(x+3h) - f(x+2h)\\ \Delta f(x+3h) &= f(x+4h) - f(x+3h)\\ \Delta f(x+mh) &= f[x+(m+1)h] - f(x+mh)\end{aligned}$$

These are the *first differences* of the function f of x. Consider $y = 2x^2 - x + 1$. Let $x = 0$, and $h = 1$. Then:

$$\begin{aligned}y_0 &= f(x) &&= 1\\ y_1 &= f(x+h) &&= 2\\ y_2 &= f(x+2h) &&= 7\\ y_3 &= f(x+3h) &&= 16\\ y_4 &= f(x+4h) &&= 29\\ y_5 &= f(x+5h) &&= 46\end{aligned}$$

The corresponding values of x and y are given in Table 5-1.

$$\begin{aligned}f(x+h) - f(x) &= 1 = \Delta f(x)\\ f(x+2h) - f(x+h) &= 5 = \Delta f(x+h)\\ f(x+3h) - f(x+2h) &= 9 = \Delta f(x+2h)\\ f(x+4h) - f(x+3h) &= 13 = \Delta f(x+3h)\\ f(x+5h) - f(x+4h) &= 17 = \Delta f(x+4h)\end{aligned}$$

Table 5-1. Values of x and y

x	y
0	1
1	2
2	7
3	16
4	29
5	46

The corresponding values of x, y, and Δf are given in Table 5-2. Note the values in the last column (Δf) lie between the values for x and y in the table 5-2.

Table 5-2. Values of Δf

x	y	Δf
0	1	
		1
1	2	
		5
2	7	
		9
3	16	
		13
4	29	
		17
5	46	

The *second differences* (or differences of order two) of $f(x)$ are:

$$\begin{aligned}\Delta^2 f(x) &= \Delta[\Delta f(x)] = \Delta[f(x+h) - f(x)]\\ &= f(x+2h) - f(x+h) - [f(x+h) - f(x)]\\ &= f(x+2h) - 2f(x+h) + f(x)\end{aligned}$$

In the same manner, we can write:

$$\Delta^2 f(x+h) = f(x+3h) - 2f(x+2h) + f(x+h)$$
$$\Delta^2 f(x+2h) = f(x+4h) - 2f(x+3h) + f(x+2h)$$

The *third differences* are the differences of the second differences, and this sequence then continues for the higher differences.

The complete list of differences for this example is given in Table 5-3.

Table 5-3. First and Second Differences of *f(x)*

x	y	Δf	$\Delta^2 f$
0	1		
		1	
1	2		4
		5	
2	7		4
		9	
3	16		4
		13	
4	29		4
		17	
5	46		

We need not obtain differences higher than n, where n is the highest exponent in the original function.

One can use differences to find the terms in a given series if it is known that differences of a specific order are constant.

For positive integral values for m, any function of $f(x)$ is given by:

$$f(x + mh) = f(x) + m\Delta f(x) + \frac{m(m-1)}{2!}\Delta^2 f(x)$$

$$+\frac{m(m-1)(m-2)}{3!}\Delta^3 f(x) + \cdots + \Delta^m f(x)$$

If $f(x)$ is a polynomial of degree n, this expression terminates at $\Delta^m f(x)$ and the nth order differences are constants.

Consider the following series:

$$5, 3, 1, 5, 21, 55, \ldots$$

Suppose $x = 0$ and $h = 1$. Then, since $x = 0$ and $h = 1$, $x + mh = m$, and we can construct the differences as in Table 5-4.

Table 5-4. Differences of *f(m)*

m	f(m)	Δf	$\Delta^2 f$	$\Delta^3 f$
0	5			
		− 2		
1	3		0	
		− 2		6
2	1		6	
		4		6
3	5		12	
		16		6
4	21		18	
		34		
5	55			

The values of $f(m)$ are given as 5, 3, 1, 5, 21, 55. Thus:

$$\begin{aligned}
\Delta f(x) &= 3 - 5 = -2\\
\Delta f(x+h) &= 1 - 3 = -2\\
\Delta f(x+2h) &= 5 - 1 = 4\\
\Delta f(x+3h) &= 21 - 5 = 16\\
\Delta f(x+4h) &= 55 - 21 = 34
\end{aligned}$$

For $x = 0$ and $m = 0$, $f(x + mh) = f(x) = f(m) = 5$.
Hence

$$\begin{aligned}
f(0+5) &= 5 + m(-2) + \frac{m(m-1)}{2!}(0)\\
&\quad + \frac{m(m-1)(m-2)}{3!}(6)\\
&= 5 - 2m + 0 + (m^2 - m)(m-2)\\
&= 5 - 2m + m^3 - m^2 - 2m^2 + 2m\\
&= 5 + m^3 - 3m^2\\
f(m) &= m^3 - 3m^2 + 5
\end{aligned}$$

Thus,

$(m = 0)$	$f(0) = 5$
$(m = 1)$	$f(1) = 1 - 3 + 5 = 3$
$(m = 2)$	$f(2) = 8 - 12 + 5 = 1$
$(m = 3)$	$f(3) = 27 - 27 + 5 = 5$
$(m = 4)$	$f(4) = 64 - 48 + 5 = 21$
$(m = 5)$	$f(5) = 125 - 75 + 5 = 55$

Consider, as an example, the determination of the common logarithm of 7.075. Now, we have:

$$\log 7.0 = 0.8451$$
$$\log 7.1 = 0.8513$$

Let $x = 7.0$ and $h = 0.1$. (Table 5-5 gives the first, second, and third differences using these values for $m = 0, \ldots, 5$.)
To obtain log 7.075 we write:

$$\begin{aligned}
x + mh &= 7.075\\
7.0 + m(0.1) &= 7.075\\
0.1m &= 0.075\\
m &= 0.75
\end{aligned}$$

Table 5-5. Calculation of log $(x + mh)$

m	x + mh	log (x + mh)	Δ	Δ^2	Δ^3
0	7.0	0.8451			
			.0062		
1	7.1	0.8513		−.0002	
			.0060		.0002
2	7.2	0.8573		.0000	
			.0060		−.0001
3	7.3	0.8633		−.0001	
			.0059		.0001
4	7.4	0.8692		.0000	
			.0059		
5	7.5	0.8751			

Hence:

$$\log 7.075 = 0.8451 + 0.75(0.0062) + \frac{0.75(-0.25)}{2}(-0.0002)$$

$$= 0.8451 + 0.004650$$

$$= 0.84975$$

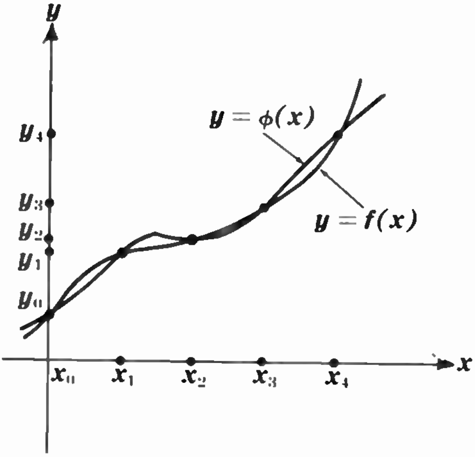


Fig. 5-3. Polynomial $\phi(x)$ approximating $f(x)$.

NEWTON'S FORMULAS

Differences can be used in many ways for interpolation. One use is in Newton's formulas for interpolation.

Consider $y = f(x)$, which, for equidistant values of x_0, x_1, x_2, x_3, . . . ,x_n, takes the values y_0, y_1, y_2, y_3, . . . , y_n (Fig. 5-3).

For a polynomial $\phi(x)$ of degree n, $\phi(x)$ can be written as:

$$\phi(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1)$$
$$+ a_3(x-x_0)(x-x_1)(x-x_2) + a_4(x-x_0)(x-x_1)(x-x_2)(x-x_3) +$$
$$\cdots + a_n(x - x_0)(x - x_1)(x - x_2)(x - x_3) \cdots (x - x_{n-1})$$

When $\Delta x = h$ one can write Newton's formula for *forward interpolation.* First, the coefficients $a_0, a_1, a_2, \ldots, a_n$ are determined so that:

$$\phi(x_0) = y_0, \phi(x_1) = y_1, \phi(x_2) = y_2, \ldots, \phi(x_n) = y_n$$

where $x_1 - x_0 = h$, $x_2 - x_0 = 2h$, and so on.
One form of Newton's formula appears as:

$$\phi(x) = y_0 + \frac{\Delta y_0}{h}(x - x_0) + \frac{\Delta^2 y_0}{2!h^2}(x - x_0)(x - x_1)$$
$$+ \frac{\Delta^3 y_0}{3!h^3}(x - x_0)(x - x_1)(x - x_2)$$
$$+ \frac{\Delta^4 y_0}{4!h^4}(x - x_0)(x - x_1)(x - x_2)(x - x_3)$$
$$+ \cdots + \frac{\Delta^n y_0}{n!h^n}(x - x_0)(x - x_1)(x - x_2) \cdots (x - x_{n-1})$$

Also, if we write $\dfrac{x - x_0}{h} = u$, then the preceding formula becomes:

$$\phi(x) = y_0 + u\Delta y_0 + \frac{u(u-1)}{2!}\Delta^2 y_0 + \frac{u(u-1)(u-2)}{3!}\Delta^3 y_0$$
$$+ \frac{u(u-1)(u-2)(u-3)}{4!}\Delta^4 y_0 + \cdots$$
$$+ \frac{u(u-1)(u-2)(u-3) \cdots (u-n+1)}{n!}\Delta^n y_0$$

These equations are known as the *forward interpolation* formulas of Newton, since they contain values of y_0 forward (to the right).

Newton's formula for *backward interpolation,* used for values to the left of a given y (backward from the y), can also be very useful. Consider the polynomial $\phi(x)$ in the form:

$$\phi(x) = a_0 + a_1(x - x_n) + a_2(x - x_n)(x - x_{n-1})$$
$$+ a_3(x - x_n)(x - x_{n-1})(x - x_{n-2})$$
$$+ a_4(x - x_n)(x - x_{n-1})(x - x_{n-2})(x - x_{n-3})$$
$$+ \cdots + a_n(x - x_n)(x - x_{n-1})(x - x_{n-2}) \cdots (x - x_1)$$

By proper substitution, the formula for backward interpolation usually appears as:

$$\phi(x) = y_n + u\Delta^1 y_n + \frac{u(u+1)}{2!}\Delta^2 y_n + \frac{u(u+1)(u+2)}{3!}\Delta^3 y_n$$

$$+ \frac{u(u+1)(u+2)(u+3)}{4!}\Delta^4 y_n + \cdots$$

$$+ \frac{u(u+1)(u+2)(u+3)\cdots(u+n-1)}{n!}\Delta^n y_n$$

LAGRANGE'S FORMULA

Another method of interpolation involves the use of Lagrange's formula, which can be expressed as:

$$y(x) = \frac{(x-x_1)(x-x_2)\cdots(x-x_n)}{(x_0-x_1)(x_0-x_2)\cdots(x_0-x_n)}y(x_0)$$

$$+ \frac{(x-x_0)(x-x_2)\cdots(x-x_n)}{(x_1-x_0)(x_1-x_2)\cdots(x_1-x_n)}y(x_1)$$

$$+ \frac{(x-x_0)(x-x_1)\cdots(x-x_n)}{(x_2-x_0)(x_2-x_1)\cdots(x_2-x_n)}y(x_2) + \cdots$$

$$+ \frac{(x-x_0)(x-x_1)\cdots(x-x_{n-1})}{(x_n-x_0)(x_n-x_1)\cdots(x_n-x_{n-1})}y(x_n)$$

This formula is useful in finding any desired value of a given function for which there are not equidistant values of the independent variable, or when one needs to locate values of the independent variable for a given value of the function.

Consider the following example. Suppose we wish to find the logarithm of 565.6, and we have the following values:

x	560.0	564.3	571.0	573.6
log x	2.74819	2.75151	2.75664	2.75861

Thus:

$$x = 565.6$$

$$x_0 = 560.0$$

$$x_1 = 564.3$$

$$x_2 = 571.0$$

$$x_3 = 573.6$$

And the differences of these values are as follows:

$$\begin{array}{lll}
x - x_1 = 1.3 & x - x_0 = 5.6 & x_2 - x_3 = -2.6 \\
x - x_2 = -5.4 & x_1 - x_0 = 4.3 & x_3 - x_0 = 13.6 \\
x - x_3 = -8.0 & x_1 - x_2 = -6.7 & x_3 - x_1 = 9.3 \\
x_0 - x_1 = -4.3 & x_1 - x_3 = -9.3 & x_3 - x_2 = 2.6 \\
x_0 - x_2 = -11.0 & x_2 - x_0 = 11.0 & \\
x_0 - x_3 = -13.6 & x_2 - x_1 = 6.7 &
\end{array}$$

$$y = \log x = \frac{(1.3)(-5.4)(-8.0)}{(-4.3)(-11.0)(-13.6)}\,2.74819$$

$$+\frac{(5.6)(-5.4)(-8.0)}{(4.3)(-6.7)(-9.3)}\,2.75151$$

$$+\frac{(5.6)(1.3)(-8.0)}{(11.0)(6.7)(-2.6)}\,2.75664$$

$$+\frac{(5.6)(1.3)(-5.4)}{(13.6)(9.3)(2.6)}\,2.75861$$

$$\log 565.6 = 2.75251$$

There are other interpolation formulas (Stirling, Bessel, and Gauss) which have different degrees of accuracy for approximating a function. For periodic functions, trigonometric formulas are often used. *Hermite's formula* for periodic functions is:

$$y(x) = \frac{\sin(x-x_1)\sin(x-x_2)\cdots\sin(x-x_n)}{\sin(x_0-x_1)\sin(x_0-x_2)\quad\sin(x_0-x_n)}\,y(x_0)$$

$$+\frac{\sin(x-x_0)\sin(x-x_2)\cdots\sin(x-x_n)}{\sin(x_1-x_0)\sin(x_1-x_2)\cdots\sin(x_1-x_n)}\,y(x_1)+\cdots$$

$$+\frac{\sin(x-x_0)\sin(x-x_1)\cdots\sin(x-x_{n-1})}{\sin(x_n-x_0)\sin(x_n-x_1)\cdots\sin(x_n-x_{n-1})}\,y(x_n)$$

NUMERICAL DIFFERENTIATION AND INTEGRATION

One can calculate the derivative of a function, given a set of values for that function. By representing the function as an interpolation formula and differentiating, one can find any desired derivative.

For example, from Stirling's interpolation formula, one can derive:

$$\left(\frac{dy}{dx}\right)_{x_0}=\frac{1}{h}\left(\frac{\Delta y_{-1}+\Delta y_0}{2}-\frac{1}{3!}\frac{\Delta^3 y_{-2}+\Delta^3 y_{-1}}{2}\right.$$

$$\left.+\frac{1\cdot 2^2}{5!}\frac{\Delta^5 y_{-3}+\Delta^5 y_{-2}}{2}+\cdots\right)$$

$$\left(\frac{d^2y}{dx^2}\right)_{x_0}=\frac{1}{h^2}\left(\frac{2\Delta^2 y_{-1}}{2!}-\frac{2}{4!}\Delta^4 y_{-2}\right.$$

$$\left.+\frac{2\cdot 2^2}{6!}\Delta^6 y_{-3}-\frac{2\cdot 2^2\cdot 3^2}{8!}\Delta^8 y_{-1}-\cdots\right)$$

$$\left(\frac{d^3y}{dx^3}\right)_{x_0}=\frac{1}{h^3}\left(\frac{\Delta^3 y_{-2}+\Delta^3 y_{-1}}{2!}-\frac{30}{5!}\frac{\Delta^5 y_{-3}+\Delta^5 y_{-2}}{2!}+\cdots\right)$$

In much the same general manner one can find the value of a definite integral $\int_a^o y(x)dx$ by replacing $y(x)$ with the interpolation formula and integrating. Although the general formulas are complex, there are certain simplifications such as *Simpson's Rule*:

$$\int_{x_0}^{x_0+nh} y(x)dx=\frac{h}{3}\sum_{k=0}^{n} c_k y(x_k)$$

where,

c_k is 1, 4, 2, ..., 2, 4, 1.

Another formula for numerical integration is Weddle's Rule:

$$\int_{x_0}^{x_0+nh} y(x)dx=\frac{3h}{10}\sum_{i=0}^{n} k_i y(x_i)$$

where for $i=0, 1, 2, \ldots, k_i$ is 1, 5, 1, 6, 1, 5, 2, 5, 1, 6, 1, 5, 2, ...

NEWTON-RAPHSON METHOD OF APPROXIMATE SOLUTION OF EQUATIONS

Consider a function $f(x)$, where the first approximation of the desired root x is x_0. The correction required (h) is such that $x = x_0 + h$. By Taylor's Series the expansion of $f(x_0 + h)$ is:

$$f(x_0 + h) = f(x_0) + hf'(x_0) + \frac{h_2}{2!}f''(x_0) + \cdots$$

Putting $f(x_0 + h) = 0$, i.e., $x_0 + h$ is a root of $f(x)$, and neglecting terms in h of the second order and higher, we have:

$$f(x_0) + hf(x_0) = 0$$

The first approximation (h_1) of h is then:

$$h_1 = \frac{f(x_0)}{f'(x_0)}$$

The next step is similar to the first:

$$x_1 = x_0 + h_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

This process may be continued to any desired approximation of x. For example, consider $f(x) = ax^2 + bx + c = 0$. Then:

$$f'(x) = 2ax + b$$

$$x_1 = \frac{ax_0^2 - c}{2ax_0 + b}$$

If the coefficients of the preceding equation are $a = 1$, $b = 0$, and $c = -u$, then the algebraic solution is $x = \sqrt{u}$. And the first numerical approximation is:

$$x_1 = \frac{1}{2}\left(x_0 + \frac{u}{x_0} \right)$$

Consider another example:

$$f(x) = 2x^3 - 7x^2 - 10x + 20$$

By graphical means, the roots are located at approximate values of -2, 1, and 4. Now:

$$f'(x) = 6x^2 - 14x - 10$$

Take $x_0 = -2$. Then:

$$\begin{aligned} h_1 &= -\frac{f(x_0)}{f'(x_0)} \\ &= -\frac{2(-8) - 7(4) + 20 + 20}{6(4) - 14(-2) - 10} \\ &= -\frac{-16 - 28 + 40}{24 + 28 - 10} = -\frac{-4}{42} \\ &= .095 \end{aligned}$$

Hence:

$$\begin{aligned} x_1 &= -2 + .095 \\ &= -1.905 \end{aligned}$$

The process can be continued as many times as required.

CHAPTER 6

Other Techniques

This chapter covers some additional techniques and methods. While it is not feasible, because of the limitations of space, to include all methods possible, iteration, matrices, and linear programming are covered in this chapter. All three are techniques used for problem solving in computers.

ITERATION

One of the most powerful tools for problem solving is iteration. For an equation such as $3x - 4 \log x = 17$, for example, one can write $x = 4 \log x + 17$ and then substitute a "good guess" for x. The equation is then evaluated, and the resulting value is used in a further substitution that results in a more nearly correct value. This method is self-correcting for errors in computation, and it is not difficult to program for a computer.

Consider a system of three equations as:

$$30x + 6y - z = 40 \qquad (1)$$

$$4x + 20y + 2z = 63 \qquad (2)$$

$$x + 2y + 40z = 50 \qquad (3)$$

Solving for each variable, we have:

$$x = \frac{1}{30}(40 - 6y + z) \qquad (4)$$

$$y = \frac{1}{20}(63 - 4x - 2z) \qquad (5)$$

$$z = \frac{1}{40}(50 - 2y - x) \qquad (6)$$

Let $y = 0$ and $z = 0$ in Equation (4). Thus:

$$x^{(1)} = \frac{40}{30} = 1.33$$

This is the *first iteration* for x, or $x^{(1)}$.

In (5), let $z = 0$ and $x = 1.33$. Then we have:

$$y^{(1)} = \frac{1}{20}(63 - 5.32) = 2.88$$

This is the first iteration for y, or $y^{(1)}$.

Let $x = 1.33$ and $y = 2.88$ in (6):

$$z^{(1)} = \frac{1}{40}(43.09) = 1.08$$

This is the first iteration for z, or $z^{(1)}$. Note that each new value is used in the following equations as soon as that value has been determined:

$$x^{(2)} = \frac{1}{30}(41.08 - 17.28) = 0.793$$

$$y^{(2)} = \frac{1}{20}(63 - 5.332) = 2.883$$

$$z^{(2)} = \frac{1}{40}(50 - 6.559) = 1.086$$

And the third iterations are:

$$x^{(3)} = \frac{1}{30}(41.086 - 17.298) = 0.793$$

$$y^{(3)} = \frac{1}{20}(63 - 5.344) = 2.883$$

$$z^{(3)} = \frac{1}{40}(50 - 6.559) = 1.086$$

The successive iterations are listed in Table 6-1.

Note that the true values are given as I_2 or I_3, and, in general, the iteration is continued until the unknown values do not change.

Table 6-1. Iterated Solutions of Equations 1-3

I	x	y	z
I_1	1.33	2.88	1.08
I_2	0.793	2.883	1.086
I_3	0.793	2.883	1.086

Iteration can also be used for certain equations. Consider an equation, such as $f(x) = 0$, that can be expressed as $x = l(x)$. A value for x_0 is determined by an approximation, which is substituted so that:

$$x^{(1)} = l(x_0)$$

Succeeding approximations are:

$$x^{(2)} = l(x^{(1)})$$
$$x^{(3)} = l(x^{(2)})$$
$$x^{(4)} = l(x^{(3)})$$
$$\cdot \quad \cdot \quad \cdot$$
$$x^{(n)} = l(x^{(n-1)})$$

Consider an example of the form $2x - \log x = 6$. This can be rewritten as:

$$y_1 = \log x, \qquad y_2 = 2x - 6$$

It is possible to plot the intersection of this pair to obtain an approximate root, or other means can be used. A plot of these equations would provide a value of about 3.25, but suppose that a first guess is incorrect, such as 4.8. This then can be used to obtain the proper value.

Starting with the approximation $x = 4.8$, we obtain:

$$x^{(1)} = \frac{1}{2}(\log 4.8 \quad + 6) = \frac{1}{2}(6.6812) = 3.3406$$

$$x^{(2)} = \frac{1}{2}(\log 3.34 + 6) = \frac{1}{2}(6.5237) = 3.2618$$

$$x^{(3)} = \frac{1}{2}(\log 3.26 + 6) = \frac{1}{2}(6.5132) = 3.2566$$

$$x^{(4)} = \frac{1}{2}(\log 3.25 + 6) = \frac{1}{2}(6.5119) = 3.2559$$

$$x^{(5)} = \frac{1}{2}(\log 3.25 + 6) = \frac{1}{2}(6.5119) = 3.2559$$

Thus, after five iterations we have $x^{(5)} = 3.2559$. But, using five-place logarithms, we obtain:

$$x^{(5)} = \frac{1}{2}(\log 3.256 \quad + 6) = \frac{1}{2}(6.51268) = 3.25634$$

$$x^{(6)} = \frac{1}{2}(\log 3.2563 + 6) = \frac{1}{2}(6.51272) = 3.25636$$

Therefore, the result is 3.2559 ($x^{(4)}$ and $x^{(5)}$), while succeeding iterations show that the value is between 3.25634 and 3.25636. Fig. 6-1 shows to four figures the approach of x to the final value.

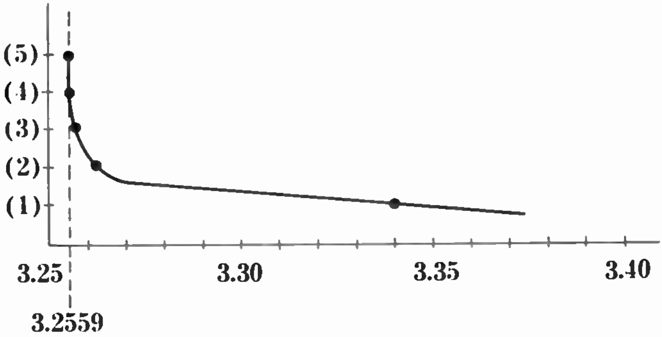


Fig. 6-1. Iteration of $x = f(x)$.

MATRICES

A matrix X is a rectangular array of numbers such as:

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix}$$

X represents an ordered array of real numbers, while m and n are integers. This is an $m \times n$ *matrix* of m rows and n columns. If $m = n$, the matrix is called a *square* matrix.

An example of a 1 × 3 matrix, which is also a row-vector, is:

$$(7, \ 0, \ 6)$$

And a 3 × 1 matrix, or a column-vector, is:

$$\begin{pmatrix} 7 \\ 0 \\ 6 \end{pmatrix}$$

Consider the example of a book publisher who has three types of books, A, B, and C. Suppose in a specific case there are 5 of kind A, 3 of kind B, 10 of kind C. Each kind takes certain hours of typesetting, illustrations, printing, and editing, as in Table 6-2.

Table 6-2. Factors in Book Publishing

Books	Type	Illustrations	Print	Edit
A	140	100	50	200
B	200	50	15	100
C	350	200	500	30

X is the row-vector (5, 3, 10), and R is a 3 × 4 matrix. We now wish to find XR.

$$XR = (5,3,10)\begin{pmatrix} 140 & 100 & 50 & 200 \\ 200 & 50 & 15 & 100 \\ 350 & 200 & 500 & 30 \end{pmatrix}$$

$$= (5(140) + 3(200) + 10(350), 5(100) + 3(50) + 10(200), 5(50) + 3(15) + 10(500), 5(200) + 3(100) + 10(30)\ )$$

$$= (4{,}800, 2{,}650, 5{,}295, 1{,}600)$$

Thus there are 4,800 hours of typesetting, 2,650 hours of illustrations, 5,295 hours of printing, and 1,600 hours of editing required.

Now, for example, it is possible to establish a cost-per-hour of $8 for typesetting, $6 for illustrating, $5 for printing, and $7 for editing. This is a matrix Z such that:

$$Z = \begin{pmatrix} 8 \\ 6 \\ 5 \\ 7 \end{pmatrix}$$

We want to find the product RZ:

$$RZ = \begin{pmatrix} 140 & 100 & 50 & 200 \\ 200 & 50 & 15 & 100 \\ 350 & 200 & 500 & 30 \end{pmatrix}\begin{pmatrix} 8 \\ 6 \\ 5 \\ 7 \end{pmatrix}$$

$$= \begin{pmatrix} 8(140) + 6(100) + 5(50) + 7(200) \\ 8(200) + 6(50) + 5(15) + 7(100) \\ 8(350) + 6(200) + 5(500) + 7(30) \end{pmatrix}$$

$$RZ = \begin{pmatrix} 3{,}370 \\ 2{,}675 \\ 6{,}710 \end{pmatrix}$$

So that the cost of the "A" books is $3,370; the "B" books cost $2,675, and the "C" books cost $6,710.

VECTORS

A vector is a special form of numerical notation for an ordered collection of numbers. A column-vector, for example, is a series of numbers written as:

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \qquad \begin{pmatrix} 1 \\ 3 \end{pmatrix} \qquad \begin{pmatrix} 9 \\ -6 \\ 2 \end{pmatrix} \qquad \begin{pmatrix} 1 \\ 6 \\ 0 \end{pmatrix}$$

The components of the vector are the individual numbers, such as x_1, x_2, x_3. Thus, the first vector above is a three-component column-vector, the last two are three-component column-vectors, and the second is a two-component column-vector. In the same manner, (1,7), (6,0), (4,–3, 2), and (0,0,6) are row-vectors.

Addition is defined on a component-addition basis. If $\mathbf{x}$ and $\mathbf{y}$ are:

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \qquad \text{and} \qquad \mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}$$

then:

$$\mathbf{x} + \mathbf{y} = \begin{pmatrix} x_1 + y_1 \\ x_2 + y_2 \\ x_3 + y_3 \end{pmatrix}$$

The addition of row-vectors is defined similarly.

Multiplication is also on a component basis. If:

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

then:

$$a\mathbf{x} = \begin{pmatrix} ax_1 \\ ax_2 \\ ax_3 \end{pmatrix}$$

Note that since a can be negative, subtraction is defined as negative addition.

The *zero-vector* is defined as:

$$\mathbf{0} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \qquad \text{or} \qquad \mathbf{0} = (0,0,0)$$

And thus $\mathbf{x} + \mathbf{0} = \mathbf{x}$.

Vectors allow group of numbers to be treated as a single unit, which is very useful in a range of problems. Here are some examples:

$$\mathbf{x} = \begin{pmatrix} 3 \\ 7 \\ 1 \end{pmatrix}, \quad \mathbf{y} = \begin{pmatrix} 2 \\ 0 \\ 1 \end{pmatrix}, \quad \mathbf{z} = \begin{pmatrix} 1 \\ 2 \\ 4 \end{pmatrix}$$

To evaluate $2\mathbf{x} + 3\mathbf{y} - \mathbf{z}$ we form the expression:

$$\begin{aligned} 2\mathbf{x} + 3\mathbf{y} - \mathbf{z} = 2 & \begin{pmatrix} 3 \\ 7 \\ 1 \end{pmatrix} + 3 \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix} - \begin{pmatrix} 1 \\ 2 \\ 4 \end{pmatrix} \\ = & \begin{pmatrix} 6 \\ 14 \\ 2 \end{pmatrix} + \begin{pmatrix} 6 \\ 6 \\ 3 \end{pmatrix} - \begin{pmatrix} 1 \\ 2 \\ 4 \end{pmatrix} \\ = & \begin{pmatrix} 2 + 3 - 4 \\ 14 + 6 - 2 \\ 6 + 6 - 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 18 \\ 11 \end{pmatrix} \end{aligned}$$

Multiplication of row-vectors and column-vectors is defined as follows. If:

$$\mathbf{x} = (x_1, x_2, x_3) \qquad \text{and} \qquad \mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}$$

then:

$$\mathbf{xy} = x_1y_1 + x_2y_2 + x_3y_3$$

Thus the product of two vectors is not itself a vector, but is instead a number. As an example, consider:

$$\mathbf{x} = (3, -2, 1), \qquad \text{and} \qquad \mathbf{y} = \begin{pmatrix} 1 \\ 4 \\ 2 \end{pmatrix}$$

Then:

$$\mathbf{xy} = 3 - 8 + 2 = -3$$

Another means of using vectors can be seen from the simple drawing in Fig. 6-2. There is a pair of axes, x_1 and x_2, as shown.

And two vectors **x** and **y**, where $\mathbf{x} = (a, b)$ and $\mathbf{y} = (c, d)$. From this point $\mathbf{x} + \mathbf{y}$ is:

$$\mathbf{x} + \mathbf{y} = (a, b) + (c, d)$$
$$\mathbf{x} + \mathbf{y} = (a + c, b + d)$$

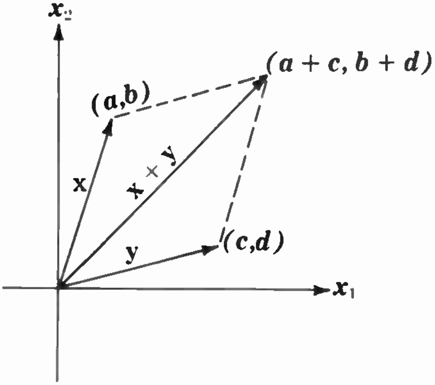


Fig. 6-2. Equation-solving by vectors.

LINEAR PROGRAMMING

Consider the linear equation $3x + 2y = 12$ as in Fig. 6-3. This equation may be plotted as a straight line, in a very simple and direct manner, since for $x = 0$, $y = 6$ and for $y = 0$, $x = 4$.

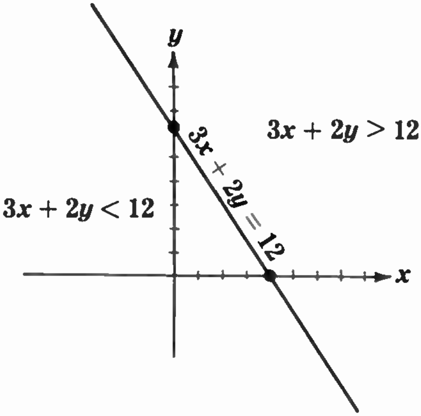


Fig. 6-3. Solving a linear equation.

Thus this straight line is the locus of all points that satisfy this equation, such as (6,0), (4,0), and (2,3). Any number of points can be found by assuming a value for x or y, and substituting it in the equation.

Now, suppose that $3x + 2y = 12$ is now written as an inequality of the form $3x + 2y < 12$. We wish to find the locus of all points that satisfy this inequality. Take a given point such as (2,3) on this line. For $x = 2$ any y less than 3 will satisfy this inequality. Any value of y greater than 3, for $x = 2$, will be above the line and will *not* satisfy the inequality. If we take $y = 3$, any $x < 3$ will again satisfy the inequality, while any $x > 3$ will not. Thus any (x, y) point below the line will satisfy the inequality $3x + 2y < 12$, and this area is known as an open half-plane whose area contains all these points.

One can show that all points above the line $3x + 2y = 12$ are in the other open half-plane corresponding to the inequality $3x + 2y > 12$. An inequality of the form $3x + 2y \leqq 12$ is made up of the open half-plane of all points below the line and the line itself; this combination is known as a closed half-plane.

Thus, for the general equation $ax + by = c$, we have:

(1) The locus of solutions is always a straight line.
(2) The inequalities $ax + by > c$ and $ax + by < c$ each have an open half-plane as its locus.
(3) The inequalities $ax + by \geqq c$ or $ax + by \leqq c$ each have a closed half-plane as its locus.

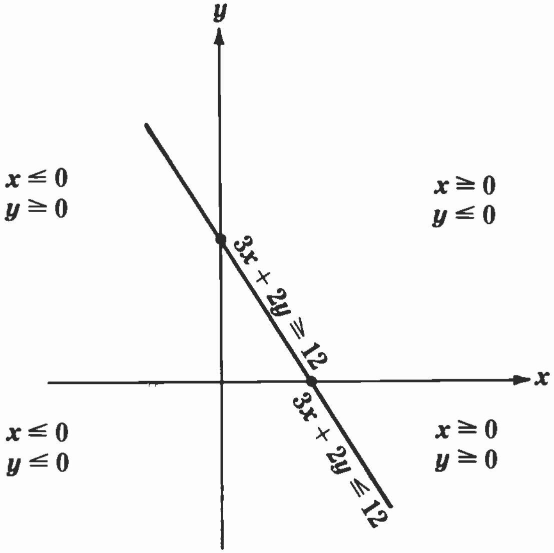


Fig. 6-4. Area defined by a system of inequalities.

Consider now a system of inequalities (Fig. 6-4):

$$x \geqq 0$$
$$y \geqq 0$$
$$3x + 2y \leqq 12$$

The locus of $x \geqq 0$ is the first and fourth quadrants, which make up the closed half-plane; the locus of $y \geqq 0$ is the first and second quadrants. Hence, only the first quadrant corresponds to both $x \geqq 0$ and $y \geqq 0$. The locus of $3x + 2y \leqq 12$ is the closed half-plane below the straight line (but including it).

From this the area that includes all points of the loci of all three inequalities is the triangle (0,6), (0,0), (4,0). This area, which is the intersection of closed half-planes, is known as a *polygonal convex set.*

Note that at any corner of the triangle, there are two and only two inequalities which become equalities at that corner. At any point on a boundary, only one of the inequalities becomes an equality; and at any interior point of the triangle there is no inequality which becomes an equality.

Consider another example. The inequalities are:

$$x > -1$$
$$y > 1$$
$$x < 2$$
$$2x + y \leqq 6$$

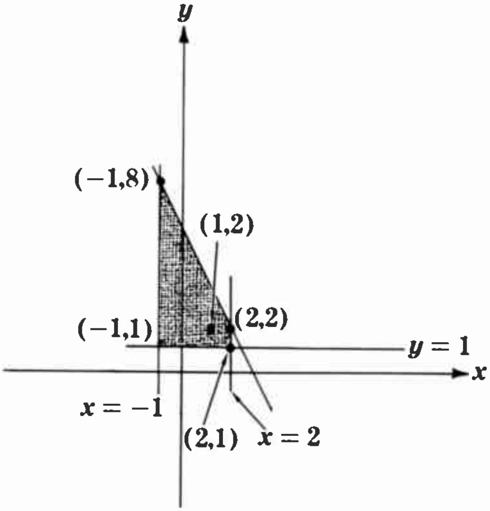


Fig. 6-5. Boundaries of *x* and *y*.

By considering $x = -1$ and $x = 2$ we establish the vertical lines as shown in Fig. 6-5. Considering $y = 1$ and $2x + y = 6$ the other boundaries are established. The inequalities for points at the vertices of the figure are shown in Table 6-3.

Table 6-3. Values at the Vertices of the Trapezoid

Point	Inequality			
	$x \geqq -1$	$y \geqq 1$	$x \leqq 2$	$2x + y \leqq 6$
(1, 8)	$x = 1$	$y = 8$	$x = 1$	$10 < 6$
(−1, 1)	$x = -1$	$y = 1$	$x = -1$	$-1 < 6$
(2, 1)	$x = 2$	$y = 1$	$x = 2$	$5 < 6$
(2, 2)	$x = 2$	$y = 2$	$x = 2$	$6 = 6$

RELAXATION METHODS

A powerful technique of problem solving, which can be used if other methods fail, is the method of *relaxation*. Consider a two-dimensional surface or membrane as in Fig. 6-6 as shown on the grid structure. Some points on this curve, as shown, are (x_1, y_2), (x_2, y_1), (x_2, y_4), (x_3, y_3), and so on. Just as a single line $(x + 3y)$ or a curve $(x^2 - 3y + xy)$ represents a set of points which can form an open loop, so the set of points above can define (enclose) an area.

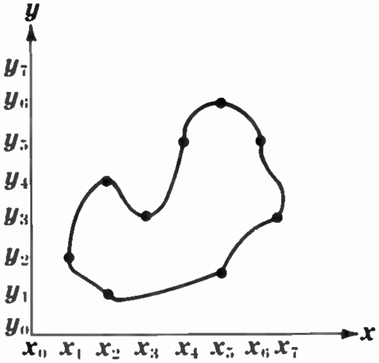


Fig. 6-6. Use of relaxation method.

Consider, as a simplification, that the curve in the figure represents a boundary of a surface with x, y, and z coordinates, but we are concerned with only the x, y relations over the surface (the z coordinates are ignored). Boundary conditions are:

$$\begin{array}{lll}
x_0y_0 = 3.0 & x_0y_1 = 17.0 & x_2y_3 = 9.0 \\
x_1y_0 = 0.0 & x_0y_2 = 10.0 & x_3y_1 = 7.5 \\
x_2y_0 = 0.0 & x_0y_3 = 5.0 & x_3y_2 = 5.0 \\
x_3y_0 = 6.0 & x_1y_3 = 7.5 & x_3y_3 = 11.0
\end{array}$$

In this example (Table 6-4) the unknown points, shown as open boxes, are to be determined. Any "guess" could be used for these values, say 5. This value, which is not correct since it is at best an approximation of the proper value, is "relaxed" toward its surrounding values. In the upper left box we use 5 for the first guess.

Table 6-4. Solving Problem by Relaxation Method

y_3	5.0	7.5	9.0	11.0
y_2	10.0	□	□	5.0
y_1	17.0	□	□	7.5
y_0	3.0	0.0	0.0	6.0
	x_0	x_1	x_2	x_3

(A)

y_3	5.0	7.5	9.0	11.0
y_2	10.0	7.3	6.5	5.0
y_1	17.0	6.7	4.8	7.5
y_0	3.0	0.0	0.0	6.0
	x_0	x_1	x_2	x_3

(B)

y_3	5.0	7.5	9.0	11.0
y_2	10.0	7.7	6.6	5.0
y_1	17.0	7.3	5.3	7.5
y_0	3.0	0.0	0.0	6.0
	x_0	x_1	x_2	x_3

(C)

y_3	5.0	7.5	9.0	11.0
y_2	10.0	7.9	6.7	5.0
y_1	17.0	7.5	5.4	7.5
y_0	3.0	0.0	0.0	6.0
	x_0	x_1	x_2	x_3

(D)

To obtain a closer value we average the surrounding numbers as 1/4 (7.5 + 10.0 + 5.0 + 5.0) = 7.3. This "better" value replaces 5, and by the same technique, the other "first-guesses" are replaced by nearer approximations. The result is the set of values shown in B. Here the central square of:

5	5
5	5

has become:

7.3	6.5
6.7	4.8

After another relaxation, the table in C results in:

7.7	6.6
7.3	5.3

And, after another relaxation (D) this is:

7.9	6.7
7.5	5.4

In this way one can arrive at a final set of values for the initial conditions.

CHAPTER 7

Other Mathematical Forms

The material in the preceding chapters has dealt with the classical mathematics of computation or numerical analysis. There are, however, other methods and techniques—indeed other forms of mathematics—that can be programmed and used on the digital computer to solve many classes of problems.

One of these is the use of symbolic logic or Boolean algebra, which is a mathematical expression of a logical system that has become exceedingly useful in circuit design, such as for control systems using complex switching arrangements. Symbolic logic is used in the programming of certain specialized types of computer language, such as COBOL (*Common Business Oriented Language*), FORTRAN (*FORmula TRANslation*), and ALGOL (a special form of algebraic language).

SYMBOLIC LOGIC

Boolean algebra derives its name from George Boole, who, in 1847, first introduced it in a paper on the mathematical analysis of logic. Until 1938, however, no practical use was found for this new algebra. Then Claude E. Shannon pointed out its application to telephone and computer switching circuits.

This extremely simple algebra provides a convenient means of representing a switching circuit without a drawing of the circuit. It also provides a means for quickly finding many circuits that will do the same thing. With a little practice a designer using Boolean algebra may quickly design "good" switching circuits. Although Boolean algebra will always suggest a good circuit, it will not always give the best circuit. (The best circuit is the circuit with the least number of components.)

Symbolic logic (Boolean algebra) is a language that can be easily programmed for a certain class of problems. The symbolic logic equation often can be manipulated into a simpler form, saving switches or gates in the implementation. The symbolic logic equation expresses the operation of a system or a circuit in terms of the basic symbolic logic operations AND, OR, and NOT. To allow a single logical function to be used for a complete system, two contemporary techniques involve NOR and NAND logic (NOT-OR and NOT-AND). The equation is developed by use of logic charts, truth tables, or simple statements of fact and can be implemented by switches and gates.

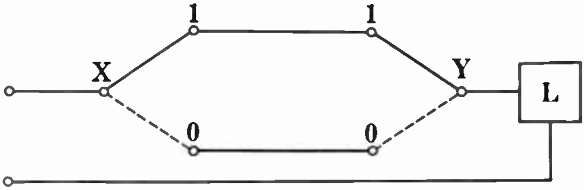


Fig. 7-1. Light controlled by 2 two-position switches in series.

If X and Y are independent variables, and A is the dependent variable, some typical equations are:

$A = X \cdot Y$ (A equals X and Y, as where X and Y are switches or relays in series.)

$A = X \vee Y$ (A equals X OR Y, as where X and Y are switches or relays in parallel.)

Consider a "1" equal to a closed switch, or a true statement, while a "0" represents an open switch, or a false statement. If A = 1, then $\overline{A}$ (NOT A) = 0.

Table 7-1. Truth Table for Light Controlled by Two Switches

X	Y	Light	Condition for Light On
0	0	1	$\overline{X} \cdot \overline{Y}$
0	1	0	
1	1	1	X•Y
1	0	0	

For example, Fig. 7-1 shows a light switch that is controlled by 2 two-position switches. Since each switch can have only two conditions, 0 and 1, there are only four possibilities, which are shown in the truth table of Table 7-1.

The symbolic-logic equation for having the light on is:

$$L = (\bar{X}\cdot\bar{Y}) \vee (X\cdot Y)$$

This is the result of two possible "light on" conditions which are $\bar{X}\cdot\bar{Y}$ as one and $X\cdot Y$ as the other.

It is desirable, of course, to reduce the symbolic logic equation to its simplest form, in order to implement it most economically. One valuable method is *De Morgan's Theorem*, which states that one can invert a statement by doing the following: (1) Change each OR into an AND; (2) change each AND into an OR; and (3) invert each term, (A, B, etc). That is, the inversion of A v B is:

$$\overline{A \vee B} = \bar{A}\cdot\bar{B}$$

This means NOT (A OR B) equals NOT A and NOT B.

By using De Morgan's Theorem, all AND-circuit gates can be changed into OR-circuit gates (or vice versa). Table 7-2 shows the truth table for the conversion of $\overline{A \text{ OR } B}$ into $\bar{A}$ AND $\bar{B}$. De Morgan's Theorem states that the expression (A v B v C) can be inverted as follows:

Table 7-2. Truth Table for $\overline{AvB} = \bar{A}\cdot\bar{B}$

A	B	A or B	$\overline{A \text{ or } B}$	$\bar{A}$	$\bar{B}$	$\bar{A}$ and $\bar{B}$
1	1	1	0	0	0	0
1	0	1	0	0	1	0
0	1	1	0	1	0	0
0	0	0	1	1	1	1

$$\overline{A \vee B \vee C} = \bar{A}\cdot\bar{B}\cdot\bar{C}$$

As the inverse of any logic expression is usually readily obtainable, De Morgan's Theorem becomes a useful method for changing the form of the expression. In the implementation of the two logic expressions, one is equivalent to the other, by De Morgan's Theorem. The proof of this equivalence is shown in the logic chart of Fig. 7-2.

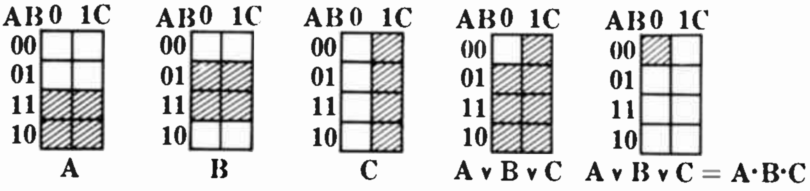


Fig. 7-2. Logic chart for $\overline{A \vee B \vee C} = \bar{A}\cdot\bar{B}\cdot\bar{C}$.

Some logic expressions can be shown to be redundant. Consider:

$$X \cdot X = X \quad (1 \text{ AND } 1 = 1;\ 0 \text{ AND } 0 = 0)$$
$$X \vee X = X \quad (1 \text{ OR } 1 = 1;\ 0 \text{ OR } 0 = 0)$$

Another redundant expression is:

$$X \cdot (X \vee Y) = X \vee (X \cdot Y) = X$$

The logic chart (Fig. 7-3) shows that this expression is the same as X. The switch analogy for this is shown in Fig. 7-4.

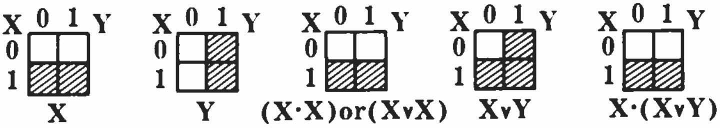


Fig. 7-3. Redundancy in logic.

Note that both $X \cdot (X \vee Y)$ and $X \vee (X \cdot Y)$ are equivalent to the simple expression X. In other words, if X is required, then other OR expressions involving X are redundant. From this it follows that:

$$X \cdot (X \vee Y \vee W) = X$$
$$X \cdot (X \vee W \vee T \vee Z) = X$$

Hence, each of these equations is redundant.

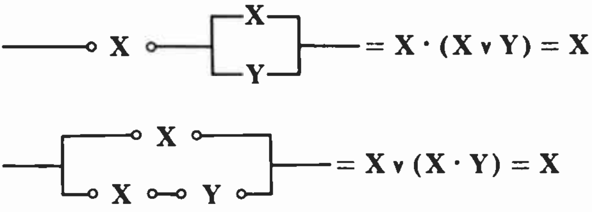


Fig. 7-4. Logically equivalent circuits.

SWITCHING CIRCUIT SIMPLIFICATION

Consider the expression:

$$X \cdot Y \cdot (X \vee Z) \cdot (Y \vee (X \cdot Z) \vee W) \cdot T$$

By a simple rearrangement of terms, this can be written:

$$Y \cdot (X \cdot (X \vee Z)) \cdot (Y \vee (X \cdot Z) \vee W) \cdot T$$

The first two terms $(X \cdot Y) \cdot (X \vee Z) = Y \cdot (X \cdot (X \vee Z))$ are equivalent to X, and the next term $(Y \vee (X \cdot Z) \vee W)$ is equivalent to Y, as shown in Fig. 7-5. Hence, this expression reduces to $X \cdot Y \cdot T$. All that this means is that if X is open (equal to 0), the entire circuit is open; this is regardless of the status of Y, Z, W, or T. The same thing applies to Y or T; each alone can open the circuit. This is something that cannot be said for W or for Z. Note that if Z is open, there can still be a complete circuit, and the same is true for W. Thus the original circuit reduces to its simplest form, $X \cdot Y \cdot T$.

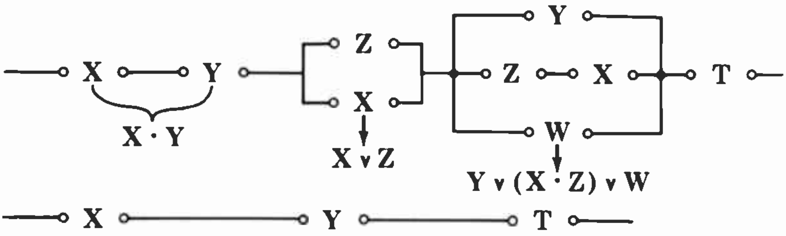


Fig. 7-5. Redundant switching circuit.

Consider the bridge circuit shown in Fig. 7-6A. The complete circuit paths possible are:

$X \cdot Y$	(1, 2, 3)
$X \cdot W \cdot S$	(1, 2, 4, 3)
$V \cdot W \cdot Y$	(1, 4, 2, 3)
$V \cdot S$	(1, 4, 3)

These are shown in Fig. 7-6B, which represents the same circuit in terms of function as that in Fig. 7-6A. One can show that this bridge circuit is the equivalent of the sum (AND's) of all complete "break-lines" characterized as *a*, *b*, *c*, or *d*. These may be represented as:

a is $Y \vee V \vee W$
b is $X \vee W \vee S$
c is $X \vee V$
d is $Y \vee S$

Their sum is:

$$(Y \vee S) \cdot (X \vee V) \cdot (X \vee W \vee S) \cdot (V \vee W \vee Y)$$

The circuit for this equation is shown in Fig. 7-5C. It is:

$$X \vee (V \cdot (W \vee S)) \cdot [Y \vee \{S \cdot (W \vee V)\}]$$

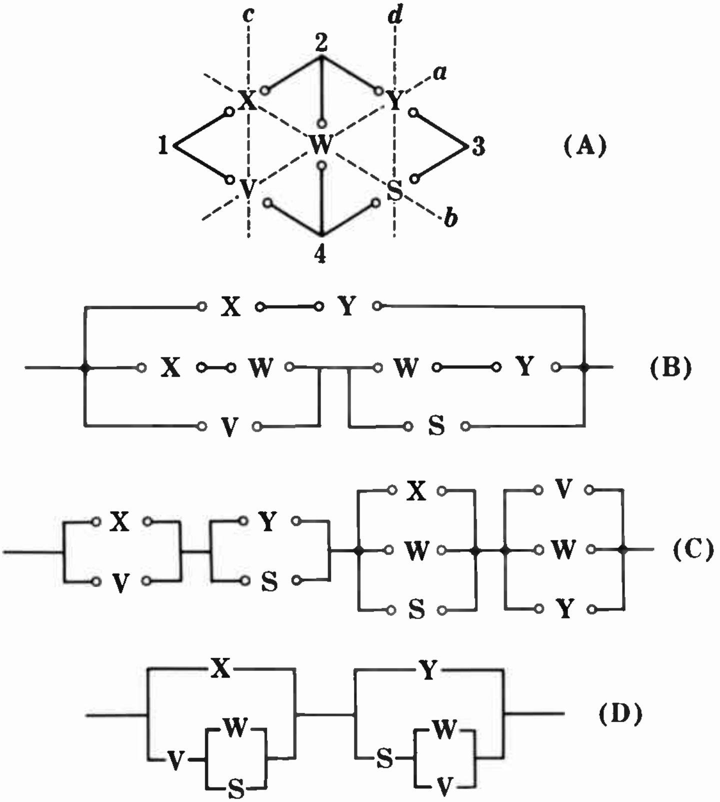


Fig. 7-6. Bridge circuit with associated logic.

This more simple version is shown in Fig. 7-6D. It can be shown that the circuits in Fig. 7-6A, B, C, and D are all equal.

LOGIC IN COMPUTER PROGRAMS

Various form of symbolic logic are used in programs for computers where "reasoning" or decisions are required.

COBOL

The COBOL (*COmmon Business Oriented Languages*) is an English-like programming language which can be used with many different types of data-processing systems. Computer programs can be written directly in machine language or indirectly in a problem-oriented language, which is a shorthand type of notation which requires translation into computer language.

A compiler or compiling system has an input language made up of abstract terms, a stored machine program (a translator) for changing or translating this input language into an output language. This output language may be in machine language that can be used directly on a specific computer, or it can be a language requiring translation. Compilers should also be considered different from interpreters which operate directly on an artificial or abstract language form to produce the computation which is the end purpose of the program.

COBOL is a shorthand for the computer's instructions. Any language is made up of words with certain meanings, and a computer language has specific words, each with a certain meaning; they can only be used with these meanings and no others. The COBOL language is derived from English; it looks like English. The programmer can work with it easily, without having to learn a long list of special symbols and codes, and the rules for using them.

For a computer to be able to interpret a COBOL sentence, the sentence must first be "translated" into the machine's language. As the COBOL system is designed, the COBOL-language program need be translated only once, and the resulting machine-language program can be used and re-used indefinitely without further translation.

A verb form in a COBOL machine language, for example, causes a certain action, and most verbs will cause some action to be taken at the time the program is executed. As an example, the verb *display* will cause specified information to be printed or otherwise displayed, and the verb *stop* will cause the computer to halt.

But there are other words and symbols besides verbs that cause action. The expression A + B − C directs that C is to be subtracted from the sum of A and B. Thus, the symbols + and − are *operators*. Operators usually specify actions or relationships without actually expressing them in verb form. There are three basic kinds of operators: arithmetic operators, relational operators, and logical operators.

The following is a list of arithmetic operators:

+	addition
−	subtraction
*	multiplication
/	division
**	exponentiation

The way in which an arithmetic expression is to be evaluated can best be specified by parentheses. Thus, the expression A*B + C might be considered ambiguous. Does the programmer mean (A*B) + C, or does he mean A*(B + C) ?If parentheses are not written to specify the order of computation, COBOL will evaluate an arithmetic expression using the following rules:

1. All exponentiation is performed first.
2. Then, multiplication and division are performed.
3. Finally, addition and subtraction are performed.
4. In each of the three preceding steps, computation starts at the left of the expression and proceeds to the right. Thus, A*B/C is computed as (A*B)/C, and A/B*C is computed as (A/B)*C.
5. When parentheses are present, computation begins with the innermost set and proceeds to the outermost. Items grouped in parentheses will be evaluated in accordance with the previous rules, and the result will then be treated as if the parentheses were removed.

Relational operators are used to express tests that are to be performed. The statement *if salary = zero* is built around the relationship implied by the equal sign, which is a relational operator. Another statement could be *if age is greater than 21 add A to B*. The words *is greater than* form a relational operator.

The relational operators in the COBOL language are: IS GREATER THAN, IS EQUAL TO, and IS LESS THAN. The words which are not underlined may be omitted, if the programmer desires, with no resulting effect on the meaning of the operator. Relational operators are combined with data to create conditional expressions and statements.

The three logical operators are AND, OR, and NOT; AND and/or OR are used when two or more tests are specified in the same expression; NOT is used to specify the negative of a condition.

Logic is also used for compound expressions. If a compound conditional expression consists of several consecutive, simple relational conditions, and if these conditions have common subjects and/or common relational operators, the common factors may be implied, instead of explicitly repeated in each condition.

For example, the compound conditional expression *married or divorced* means that either or both of these two conditions must be true for the expression as a whole to be true. The expression A AND (B OR C) OR D means one of the following must be true for the expression to be true:

1. D must be true, or
2. A must be true and either B or C (or both) must be true, or
3. Both the above must be true.

COBOL allows the four operations of addition, subtraction, multiplication, and division. For example, consider a case of writing: ADD RECEIPTS TO STOCK-ON-HAND. SUBTRACT SHIPMENTS FROM STOCK ON HAND. COMPUTE STOCK-ON-HAND VALUE. This is clear in meaning to both the computer processor and the casual reader. It is also possible to write this as: COMPUTE STOCK-VALUE = UNIT-PRICE * (STOCK-ON-HAND + RECEIPTS − SHIPMENTS. This will result in the following steps:

1. Take the stock-on-hand (number) and add the receipts.
2. From this number subtract the shipments.
3. Multiply (*) this number of units by the unit price to obtain the stock value.

This is an example of a part of a COBOL program.

ALGOL

ALGOL is a language used for programming algebraic computations much as COBOL is used for problems with business orientation. ALGOL stands for *ALGOrithmic Language*. It has been accepted as a common international language for use with algebraic problems. As with other specialized languages it is not possible to explain its use unless certain general aspects have been established.

A typical use of ALGOL would be to evaluate an expression such as $15x^3 - 7x^2 + 2x - 5$. The range of x may be a series of values such as 3, 10, 15, 41, and 62. Each value is substituted for x, and the expression is worked out. The results are printed out or punched out on cards or punched tape. ALGOL permits a direct means of writing out the required program.

There are various types of ALGOL programs. Those that are intended just for the human reader and not for use on a computer are called the *publication form* of the language. There are various machine (computer) programs by which ALGOL is turned into a program that can solve the specific problem. For simple equations or relationships these steps seem trivial. Actually, for complex programs ALGOL not only saves time, but it also allows the program to be determined without regard to the actual computer's construction and features.

The beginning is the basic letter, number, or symbol that is the character. The characters are of different types; the letters are upper-case A through Z and lower-case a through z. The digits are 0 through 9. The logical values are T for true, and F for false; these are the same as letters T and F in the series of letters, but they are told apart by their use. The typical arithmetic operators are as below:

+	plus (addition)
–	minus (subtraction)
×	multiplication
/	division
	exponentiation (B3 means B^3)

These operators can be used with letters to produce meaningful relations such as a + b, or d – f, or others.

Certain other ALGOL words are treated like symbols and have specific meanings. Words are made up of characters and, in turn, words make up expressions.

An automatic programming statement is like an English sentence or an instruction in computer language. Statements are used in the ALGOL program, which is often a succession of related statements. For example, "read (*a*, *b*);" means read the two numbers available as the input and assign these values to variables *a*, *b*. Thus "read (*a*, *b*);" is a statement, and all statements are separated by semicolons (;), as shown. The symbol : = is a single symbol.

The use of arithmetic operators such as +, –, × and others in expressions requires care, for more than one interpretation is possible in some cases. Here are some examples and their ALGOL meaning:

$a + b - c$	*means*	$(a + b) - c$
$a - b + c$	*means*	$(a - b) + c$
$a + b \quad c + d$	*means*	$a + (b \quad c) + d$
$-a \quad b$	*means*	$-(a \quad b)$
$-a - b - c$	*means*	$((-a) - b) - c$

FORTRAN

FORTRAN (*FORmula TRANslation*) is a mathematical language for computer programming. Using symbols, a program, which is coded in FORTRAN, might be, for example, D = A + X – Y. The number of instructions required for the complete solution of a problem may be a few hundred or many thousands, depending on the problem. The computer refers to them one after another, or it can be instructed to repeat, modify or skip over certain instructions, depending on immediate results or circumstances. However, such circumstances must be anticipated, and appropriate instructions included in the program.

Each FORTRAN program is composed of a number of *statements*. Each statement deals with one aspect of the problem; it may cause data to be fed into the computer, calculations to be performed, decisions to be made, results to be printed, etc.

Some statements written by the programmer do not cause specific computer action, but rather provide information to the processor. Some examples of FORTRAN statements and their effects are:

READ 1,A — This causes the computer to read an IBM card and handle the data on it in such a way that if the card read has the number 106.7 on it, then A will have the value 106.7.

C = 3.*A — The asterisk (*) indicates multiplication; so this statement means multiply A by 3.0, and set C equal to the result.

To find the roots of a quadratic equation, the computer must be told how to find the roots:

$$root = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A}$$

Since there are two roots for the quadratic equation, the computer must be instructed how to find each root separately. Thus, by using the formula, the computer may be instructed as follows:

$$root\ 1 = (-B+\mathrm{SQRTF}\ (B{*}{*}2-4.\ {*}A{*}C))/(2.{*}A)$$

$$root\ 2 = (-B-\mathrm{SQRTF}\ (B{*}{*}2-4.\ {*}A{*}C))/(2.{*}A)$$

Parentheses may be used to specify the order of operation in an expression. Where parentheses are omitted, the order is taken to be from left to right as follows:

**	exponentiation
* and /	multiplication and division
+ and −	addition and subtraction

APPENDIX I

Glossary

accumulator—A portion of the digital computer which can store a number and, upon the proper signal, add a second number to its contents, so that the sum is formed. The accumulator can also be used for storage as well as for addition.

accuracy—This is a measure of the freedom from error. This is not the same as precision, since precision is the degree of fineness, which is a different concept. A five-place logarithm table, for example, can be just as accurate as a six-place table, but the six-place table is more precise than the five-place.

addend—This is a quantity or number to be added to a second number, the augend, to produce a result known as the *sum*.

address—A name or label designates a register used for storage of a number.

analog computer—A type of computer, often electronic, where numerical quantities are represented by electrical variables, and problem solutions are obtained by manipulating these variables.

AND—A logical connection, as in the statement A AND B, which means that the statement is true if, and only if, A is true and B is true simultaneously.

arithmetic operation—Various manipulations of numerical quantities, which include the fundamental operations of addition, subtraction, multiplication, and division.

arithmetic shift—The multiplication or division of a quantity by the power of the base notation. For example, since 1011_2 represents 11 in binary notation, the result of two shifts to the left is 101100_2, which represents 44 in binary notation.

base—The radix or the numbering base. Also a number used to define a positional notation of numbering representation.

binary—A numbering system based on two values or a numbering system to the base of 2.

binary digit—A digit in the binary scale of notation. This digit may be a 0 (zero) or a 1 (one). Abbreviated as *Bit.*

binary notation—The writing of numbers in the scale of two. The first dozen numbers, zero to eleven, are written, 0, 1, 10, 11, 100, 101, 110, 111, 1000, 1001, 1010, 1011. The positions of the digits represent powers of 2; thus, 1010 means 1 times 2^3, or 8, plus 0 times 2^2, or 0, plus 1 times 2^1, or 2, and 0 times 2^0, or 0; thus, 1010 is equal to 8 +2, or 10.

binary number—A digit or group of digits representing a number to the base 2 and having one of two possible values that are 0 and 1.

binary point—In a binary number, the point marks the place between the integral powers of two and the fractional powers of two, analogous to the decimal point in the decimal system. Thus 10.101_2 means 2 + 1/2 + 1/8.

binary to decimal conversion—The mathematical process of converting a number written in binary notation to the equivalent number written in decimal notation.

bit—A binary digit.

carry—A process of bringing forward. The carry digit or the digit which is to be added to the next higher column, or a special condition which occurs when the sum of two digits in a single column is equal to or greater than the numbering base.

code—As used in this book, a system of symbols, usually numerical, used to represent various values in a system of weighted values.

code, binary decimal—A coding system in which each individual decimal digit (rather than the entire number) is represented by a binary or two-valued code.

code, excess 3 (XS-3)—One of the series of numerical codes where each of the decimal digits is represented as the binary number plus 3 in a coded decimal notation. For example, 0 is binary 3, 1 is binary 4, and 7 is binary 10.

code, gray—A special type of coded or reflected binary system where there is a change in only one digit in going from one decimal number to the next higher decimal number.

complement—A quantity derived from a given number in such a way so that the sum of the two individual numbers is equal to the next higher place to the particular radix. To the base 10, the complement of 7 is 3; and to the base 10, the complement of 8 is 12.

conversion—A method of changing the representation of information from one system to another, as in converting a number from the binary system to the decimal system.

data—Information, particularly as taken in, operated on, or put out by a computer or other machine for handling information.

decimal code, binary—A type of decimal notation where the individual decimal digits are represented by means of a binary code rather than the binary representation of the entire decimal number.

decimal digit—One of the symbols 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 when used in numbering in the scale of ten.

digit—Considered as one of n symbols, having an integral value and ranging from 0 to n-1, such as the decimal digits 0, 1, 2, 3, and so forth.

digit, binary—A 1 or 0 in the binary, or two-valued, numbering system.

dividend—A number, which when divided by another number known as the divisor, produces a result known as the quotient.

fixed-point—A system of arithmetic or notation in which all of the numerical quantities are represented by a number of digits that are predetermined with the radix point at some fixed location.

gate—An electronic circuit designed to produce an output under specific conditions, such as the AND gate, which requires both inputs to be present; and the OR gate, which requires either of two inputs to be present to produce an output.

instruction—Considered as a group of characters which are used to divide an operation, such as addition, and the address or addresses where the information specified is to be obtained.

msd—Most significant digit.

multiplicand—A quantity or number that is to be multiplied by a second number known as the multiplier to produce the result known as the product.

octal—Relating to a numbering system of radix 8.

operand—Any one of the quantities entering into or arising from an operation. An operand may be a result, a parameter, or a storage location.

operator—Usually a mathematical symbol representing a particular mathematical process to be performed.

OR circuit—A circuit which has a number of input lines and one output line, so that whenever a signal is present on one or more of the input lines, a signal is present on the output line.

parity—The condition of a number being either even or odd.

point—In a scale of numeric notation, the position designated by a dot that marks the separation between the integral and fractional parts of the number.

precision—As contrasted to accuracy, the degree of exactness to which it is possible to measure a quantity; the precision is often related to the number of significant digits representing a given value.

program—A precise sequence of coded instructions directing a computer to solve a given type of problem.

programmer—A person who prepares sequences of instructions for computer.

quantity—A positive or negative real number in the mathematical sense.

radix—One of the schemes for representing numbers, characterized by the arrangement in sequence of digits with the understanding that successive digits are multiplied by coefficients of successive powers of an integer called the base of the number system. The base of the decimal system is 10, and the base of the binary system is 2.

scale (scaling)—A technique used to alter or change the measure of units so that all variables are expressed within a certain range of magnitude.

self-checking code—A code that utilizes certain expressions so that an error produces a "forbidden" combination, thus stopping the computer or producing an alarm.

significant digits—If the digits of a number are ranked according to their ascending higher-powers of the base, then the significant digits are those ranging from the highest-power digit (different from zero) and ending with the lowest-power digit.

word—A group of characters as used in the computer so that the group is treated as and acts as a single unit.

APPENDIX II

Fractional Decimal-Octal Conversions

$(n)_{10}$	$(n)_8$	$(n \times 10^{-3})_8$	$(n \times 10^{-6})_8$
0.000	0.00000 00000 0	0.00000 00000 0	0.00000 00000 0
.001	.00040 61115 7	.00000 02061 6	.00000 00001 1
.002	.00101 42233 5	.00000 04143 4	.00000 00002 1
.003	.00142 23351 4	.00000 06225 2	.00000 00003 2
.004	.00203 04467 2	.00000 10307 0	.00000 00004 2
0.005	0.00243 65605 1	0.00000 12370 6	0.00000 00005 3
.006	.00304 46723 0	.00000 14452 4	.00000 00006 4
.007	.00345 30040 6	.00000 16534 2	.00000 00007 4
.008	.00406 11156 5	.00000 20615 7	.00000 00010 5
.009	.00446 72274 3	.00000 22677 5	.00000 00011 5
0.010	0.00507 53412 2	0.00000 24761 3	0.00000 00012 6
.011	.00550 34530 1	.00000 27043 1	.00000 00013 6
.012	.00611 15645 7	.00000 31124 7	.00000 00014 7
.013	.00651 76763 6	.00000 33206 5	.00000 00016 0
.014	.00712 60101 4	.00000 35270 3	.00000 00017 0
0.015	0.00753 41217 3	0.00000 37352 1	0.00000 00020 1
.016	.01014 22335 1	.00000 41433 7	.00000 00021 1
.017	.01055 03453 0	.00000 43515 5	.00000 00022 2
.018	.01115 64570 7	.00000 45577 3	.00000 00023 3
.019	.01156 45706 5	.00000 47661 1	.00000 00024 3
0.020	0.01217 27024 4	0.00000 51742 7	0.00000 00025 4
.021	.01260 10142 2	.00000 54024 5	.00000 00026 4
.022	.01320 71260 1	.00000 56106 3	.00000 00027 5
.023	.01361 52376 0	.00000 60170 0	.00000 00030 6
.024	.01422 33513 6	.00000 62251 6	.00000 00031 6
0.025	0.01463 14631 5	0.00000 64333 4	0.00000 00032 7
.026	.01523 75747 3	.00000 66415 2	.00000 00033 7
.027	.01564 57065 2	.00000 70477 0	.00000 00035 0
.028	.01625 40203 1	.00000 72560 6	.00000 00036 1
.029	.01666 21320 7	.00000 74642 4	.00000 00037 1
0.030	0.01727 02436 6	0.00000 76724 2	0.00000 00040 2
.031	.01767 63554 4	.00001 01006 0	.00000 00041 2
.032	.02030 44672 3	.00001 03067 6	.00000 00042 3
.033	.02071 26010 2	.00001 05151 4	.00000 00043 3
.034	.02132 07126 0	.00001 07233 2	.00000 00044 4
0.035	0.02172 70243 7	0.00001 11315 0	0.00000 00045 5
.036	.02233 51361 5	.00001 13376 6	.00000 00046 5
.037	.02274 32477 4	.00001 15460 4	.00000 00047 6
.038	.02335 13615 2	.00001 17542 2	.00000 00050 6
.039	.02375 74733 1	.00001 21623 7	.00000 00051 7
0.040	0.02436 56051 0	0.00001 23705 5	0.00000 00053 0
.041	.02477 37166 6	.00001 25767 3	.00000 00054 0
.042	.02540 20304 5	.00001 30051 1	.00000 00055 1
.043	.02601 01422 3	.00001 32132 7	.00000 00056 1
.044	.02641 62540 2	.00001 34214 5	.00000 00057 2

(Courtesy Computer Control Co.)

$(n)_{10}$	$(n)_8$	$(n \times 10^{-3})_8$	$(n \times 10^{-6})_8$
0.045	0.02702 43656 1	0.00001 36276 3	0.00000 00060 3
.046	.02743 24773 7	.00001 40360 1	.00000 00061 3
.047	.03004 06111 6	.00001 42441 7	.00000 00062 4
.048	.03044 67227 4	.00001 44523 5	.00000 00063 4
.049	.03105 50345 3	.00001 46605 3	.00000 00064 5
0.050	0.03146 31463 2	0.00001 50667 1	0.00000 00065 5
.051	.03207 12601 0	.00001 52750 7	.00000 00066 6
.052	.03247 73716 7	.00001 55032 5	.00000 00067 7
.053	.03310 55034 5	.00001 57114 3	.00000 00070 7
.054	.03351 36152 4	.00001 61176 0	.00000 00072 0
0.055	0.03412 17270 3	0.00001 63257 6	0.00000 00073 0
.056	.03453 00406 1	.00001 65341 4	.00000 00074 1
.057	.03513 61524 0	.00001 67423 2	.00000 00075 2
.058	.03554 42641 6	.00001 71505 0	.00000 00076 2
.059	.03615 23757 5	.00001 73566 6	.00000 00077 3
0.060	0.03656 05075 4	0.00001 75650 4	0.00000 00100 3
.061	.03716 66213 2	.00001 77732 2	.00000 00101 4
.062	.03757 47331 1	.00002 02014 0	.00000 00102 5
.063	.04020 30446 7	.00002 04075 6	.00000 00103 5
.064	.04061 11564 6	.00002 06157 4	.00000 00104 6
0.065	0.04121 72702 4	0.00002 10241 2	0.00000 00105 6
.066	.04162 54020 3	.00002 12323 0	.00000 00106 7
.067	.04223 35136 2	.00002 14404 6	.00000 00110 0
.068	.04264 16254 0	.00002 16466 4	.00000 00111 0
.069	.04324 77371 7	.00002 20550 1	.00000 00112 1
0.070	0.04365 60507 5	0.00002 22631 7	0.00000 00113 1
.071	.04426 41625 4	.00002 24713 5	.00000 00114 2
.072	.04467 22743 3	.00002 26775 3	.00000 00115 2
.073	.04530 04061 1	.00002 31057 1	.00000 00116 3
.074	.04570 65177 0	.00002 33140 7	.00000 00117 4
0.075	0.04631 46314 6	0.00002 35222 5	0.00000 00120 4
.076	.04672 27432 5	.00002 37304 3	.00000 00121 5
.077	.04733 10550 4	.00002 41366 1	.00000 00122 5
.078	.04773 71666 2	.00002 43447 7	.00000 00123 6
.079	.05034 53004 1	.00002 45531 5	.00000 00124 7
0.080	0.05075 34121 7	0.00002 47613 3	0.00000 00125 7
.081	.05136 15237 5	.00002 51675 1	.00000 00127 0
.082	.05176 76355 5	.00002 53756 7	.00000 00130 0
.083	.05237 57473 3	.00002 56040 5	.00000 00131 1
.084	.05300 40611 2	.00002 60122 3	.00000 00132 2
0.085	0.05341 21727 0	0.00002 62204 0	0.00000 00133 2
.086	.05402 03044 7	.00002 64265 6	.00000 00134 3
.087	.05442 64162 6	.00002 66347 4	.00000 00135 3
.088	.05503 45300 4	.00002 70431 2	.00000 00136 4
.089	.05544 26416 3	.00002 72513 0	.00000 00137 4
0.090	0.05605 07534 1	0.00002 74574 6	0.00000 00140 5
.091	.05645 70652 0	.00002 76656 4	.00000 00141 6
.092	.05706 51767 6	.00003 00740 2	.00000 00142 6
.093	.05747 33105 5	.00003 03022 0	.00000 00143 7
.094	.06010 14223 4	.00003 05103 6	.00000 00144 7

$(n)_{10}$	$(n)_8$	$(n \times 10^{-3})_8$	$(n \times 10^{-6})_8$
0.095	0.06050 75341 2	0.00003 07165 4	0.00000 00146 0
.096	.06111 56457 1	.00003 11247 2	.00000 00147 1
.097	.06152 37574 7	.00003 13331 0	.00000 00150 1
.098	.06213 20712 6	.00003 15412 6	.00000 00151 2
.099	.06254 02030 5	.00003 17474 4	.00000 00152 2
0.100	0.06314 63146 3	0.00003 21556 1	0.00000 00153 3
.101	.06355 44264 2	.00003 23637 7	.00000 00154 4
.102	.06416 25402 0	.00003 25721 5	.00000 00155 4
.103	.06457 06517 7	.00003 30003 3	.00000 00156 5
.104	.06517 67635 6	.00003 32065 1	.00000 00157 5
0.105	0.06560 50753 4	0.00003 34146 7	0.00000 00160 6
.106	.06621 32071 3	.00003 36230 5	.00000 00161 7
.107	.06662 13207 1	.00003 40312 3	.00000 00162 7
.108	.06722 74325 0	.00003 42374 1	.00000 00164 0
.109	.06763 55442 7	.00003 44455 7	.00000 00165 0
0.110	0.07024 36560 5	0.00003 46537 5	0.00000 00166 1
.111	.07065 17676 4	.00003 50621 3	.00000 00167 1
.112	.07126 01014 2	.00003 52703 1	.00000 00170 2
.113	.07166 62132 1	.00003 54764 7	.00000 00171 3
.114	.07227 43247 7	.00003 57046 5	.00000 00172 3
0.115	0.07270 24365 6	0.00003 61130 2	0.00000 00173 4
.116	.07331 05503 5	.00003 63212 0	.00000 00174 4
.117	.07371 66621 3	.00003 65273 6	.00000 00175 5
.118	.07432 47737 2	.00003 67355 4	.00000 00176 6
.119	.07473 31055 0	.00003 71437 2	.00000 00177 6
0.120	0.07534 12172 7	0.00003 73521 0	0.00000 00200 7
.121	.07574 73310 6	.00003 75602 6	.00000 00201 7
.122	.07635 54426 4	.00003 77664 4	.00000 00203 0
.123	.07676 35544 3	.00004 01746 2	.00000 00204 1
.124	.07737 16662 1	.00004 04030 0	.00000 00205 1
0.125	0.10000 00000 0	0.00004 06111 6	0.00000 00206 2
.126	.10040 61115 7	.00004 10173 4	.00000 00207 2
.127	.10101 42233 5	.00004 12255 2	.00000 00210 3
.128	.10142 23351 4	.00004 14337 0	.00000 00211 3
.129	.10203 04467 2	.00004 16420 6	.00000 00212 4
0.130	0.10243 65605 1	0.00004 20502 3	0.00000 00213 5
.131	.10304 46723 0	.00004 22564 1	.00000 00214 5
.132	.10345 30040 6	.00004 24645 7	.00000 00215 6
.133	.10406 11156 5	.00004 26727 5	.00000 00216 6
.134	.10446 72274 3	.00004 31011 3	.00000 00217 7
0.135	0.10507 53412 2	0.00004 33073 1	0.00000 00221 0
.136	.10550 34530 1	.00004 35154 7	.00000 00222 0
.137	.10611 15645 7	.00004 37236 5	.00000 00223 1
.138	.10651 76763 6	.00004 41320 3	.00000 00224 1
.139	.10712 60101 4	.00004 43402 1	.00000 00225 2
0.140	0.10753 41217 3	0.00004 45463 7	0.00000 00226 3
.141	.11014 22335 1	.00004 47545 5	.00000 00227 3
.142	.11055 03453 0	.00004 51627 3	.00000 00230 4
.143	.11115 64570 7	.00004 53711 1	.00000 00231 4
.144	.11156 45706 5	.00004 55772 7	.00000 00232 5

$(n)_{10}$	$(n)_8$	$(n \times 10^{-3})_8$	$(n \times 10^{-6})_8$
0.145	0.11217 27024 4	0.00004 60054 5	0.00000 00233 6
.146	.11260 10142 2	.00004 62136 2	.00000 00234 6
.147	.11320 71260 1	.00004 64220 0	.00000 00235 7
.148	.11361 52376 0	.00004 66301 6	.00000 00236 7
.149	.11422 33513 6	.00004 70363 4	.00000 00240 0
0.150	0.11463 14631 5	0.00004 72445 2	0.00000 00241 0
.151	.11523 75747 3	.00004 74527 0	.00000 00242 1
.152	.11564 57065 2	.00004 76610 6	.00000 00243 2
.153	.11625 40203 1	.00005 00672 4	.00000 00244 2
.154	.11666 21320 7	.00005 02754 2	.00000 00245 3
0.155	0.11727 02436 6	0.00005 05036 0	0.00000 00246 3
.156	.11767 63554 4	.00005 07117 6	.00000 00247 4
.157	.12030 44672 3	.00005 11201 4	.00000 00250 5
.158	.12071 26010 2	.00005 13263 2	.00000 00251 5
.159	.12132 07126 0	.00005 15345 0	.00000 00252 6
0.160	0.12172 70243 7	0.00005 17426 6	0.00000 00253 6
.161	.12233 51361 5	.00005 21510 3	.00000 00254 7
.162	.12274 32477 4	.00005 23572 1	.00000 00256 0
.163	.12335 13615 2	.00005 25653 7	.00000 00257 0
.164	.12375 74733 1	.00005 27735 5	.00000 00260 1
0.165	0.12436 56051 0	0.00005 32017 3	0.00000 00261 1
.166	.12477 37166 6	.00005 34101 1	.00000 00262 2
.167	.12540 20304 5	.00005 36162 7	.00000 00263 3
.168	.12601 01422 3	.00005 40244 5	.00000 00264 3
.169	.12641 62540 2	.00005 42326 3	.00000 00265 4
0.170	0.12702 43656 1	0.00005 44410 1	0.00000 00266 4
.171	.12743 24773 7	.00005 46471 7	.00000 00267 5
.172	.13004 06111 6	.00005 50553 5	.00000 00270 5
.173	.13044 67227 4	.00005 52635 3	.00000 00271 6
.174	.13105 50345 3	.00005 54717 1	.00000 00272 7
0.175	0.13146 31463 2	0.00005 57000 7	0.00000 00273 7
.176	.13207 12601 0	.00005 61062 4	.00000 00275 0
.177	.13247 73716 7	.00005 63144 2	.00000 00276 0
.178	.13310 55034 5	.00005 65226 0	.00000 00277 1
.179	.13351 36152 4	.00005 67307 6	.00000 00300 2
0.180	0.13412 17270 3	0.00005 71371 4	0.00000 00301 2
.181	.13453 00406 1	.00005 73453 2	.00000 00302 3
.182	.13513 61524 0	.00005 75535 0	.00000 00303 3
.183	.13554 42641 6	.00005 77616 6	.00000 00304 4
.184	.13615 23757 5	.00006 01700 4	.00000 00305 5
0.185	0.13656 05075 4	0.00006 03762 2	0.00000 00306 5
.186	.13716 66213 2	.00006 06044 0	.00000 00307 6
.187	.13757 47331 1	.00006 10125 6	.00000 00310 6
.188	.14020 30446 7	.00006 12207 4	.00000 00311 7
.189	.14061 11564 6	.00006 14271 2	.00000 00312 7
0.190	0.14121 72702 4	0.00006 16353 0	0.00000 00314 0
.191	.14162 54020 3	.00006 20434 6	.00000 00315 1
.192	.14223 35136 2	.00006 22516 3	.00000 00316 1
.193	.14264 16254 0	.00006 24600 1	.00000 00317 2
.194	.14324 77371 7	.00006 26661 7	.00000 00320 2

$(n)_{10}$	$(n)_8$	$(n \times 10^{-3})_8$	$(n \times 10^{-6})_8$
0.195	0.14365 60507 5	0.00006 30743 5	0.00000 00321 3
.196	.14426 41625 4	.00006 33025 3	.00000 00322 4
.197	.14467 22743 3	.00006 35107 1	.00000 00323 4
.198	.14530 04061 1	.00006 37170 7	.00000 00324 5
.199	.14570 65177 0	.00006 41252 5	.00000 00325 5
0.200	0.14631 46314 6	0.00006 43334 3	0.00000 00326 6
.201	.14672 27432 5	.00006 45416 1	.00000 00327 7
.202	.14733 10550 4	.00006 47477 7	.00000 00330 7
.203	.14773 71666 2	.00006 51561 5	.00000 00332 0
.204	.15034 53004 1	.00006 53643 3	.00000 00333 0
0.205	0.15075 34121 7	0.00006 55725 1	0.00000 00334 1
.206	.15136 15237 6	.00006 60006 7	.00000 00335 2
.207	.15176 76355 5	.00006 62070 4	.00000 00336 2
.208	.15237 57473 3	.00006 64152 2	.00000 00337 3
.209	.15300 40611 2	.00006 66234 0	.00000 00340 3
0.210	0.15341 21727 0	0.00006 70315 6	0.00000 00341 4
.211	.15402 03044 7	.00006 72377 4	.00000 00342 4
.212	.15442 64162 6	.00006 74461 2	.00000 00343 5
.213	.15503 45300 4	.00006 76543 0	.00000 00344 6
.214	.15544 26416 3	.00007 00624 6	.00000 00345 6
0.215	0.15605 07534 1	0.00007 02706 4	0.00000 00346 7
.216	.15645 70652 0	.00007 04770 2	.00000 00347 7
.217	.15706 51767 6	.00007 07052 0	.00000 00351 0
.218	.15747 33105 5	.00007 11133 6	.00000 00352 1
.219	.16010 14223 4	.00007 13215 4	.00000 00353 1
0.220	0.16050 75341 2	0.00007 15277 2	0.00000 00354 2
.221	.16111 56457 1	.00007 17361 0	.00000 00355 2
.222	.16152 37574 7	.00007 21442 5	.00000 00356 3
.223	.16213 20712 6	.00007 23524 3	.00000 00357 4
.224	.16254 02030 5	.00007 25606 1	.00000 00360 4
0.225	0.16314 63146 3	0.00007 27667 7	0.00000 00361 5
.226	.16355 44264 2	.00007 31751 5	.00000 00362 5
.227	.16416 25402 0	.00007 34033 3	.00000 00363 6
.228	.16457 06517 7	.00007 36115 1	.00000 00364 6
.229	.16517 67635 6	.00007 40176 7	.00000 00365 7
0.230	0.16560 50753 4	0.00007 42260 5	0.00000 00367 0
.231	.16621 32071 3	.00007 44342 3	.00000 00370 0
.232	.16662 13207 1	.00007 46424 1	.00000 00371 1
.233	.16722 74325 0	.00007 50505 7	.00000 00372 1
.234	.16763 55442 7	.00007 52567 5	.00000 00373 2
0.235	0.17024 36560 5	0.00007 54651 3	0.00000 00374 3
.236	.17065 17676 4	.00007 56733 1	.00000 00375 3
.237	.17126 01014 2	.00007 61014 6	.00000 00376 4
.238	.17166 62132 1	.00007 63076 4	.00000 00377 4
.239	.17227 43247 7	.00007 65160 2	.00000 00400 5
0.240	0.17270 24365 6	0.00007 67242 0	0.00000 00401 6
.241	.17331 05503 5	.00007 71323 6	.00000 00402 6
.242	.17371 66621 3	.00007 73405 4	.00000 00403 7
.243	.17432 47737 2	.00007 75467 2	.00000 00404 7
.244	.17473 31055 0	.00007 77551 0	.00000 00406 0

$(n)_{10}$	$(n)_8$	$(n \times 10^{-3})_8$	$(n \times 10^{-6})_8$
0.245	0.17534 12172 7	0.00010 01632 6	0.00000 00407 1
.246	.17574 73310 6	.00010 03714 4	.00000 00410 1
.247	.17635 54426 4	.00010 05776 2	.00000 00411 2
.248	.17676 35544 3	.00010 10060 0	.00000 00412 2
.249	.17737 16662 1	.00010 12141 6	.00000 00413 3
0.250	0.20000 00000 0	0.00010 14223 4	0.00000 00414 3
.251	.20040 61115 7	.00010 16305 2	.00000 00415 4
.252	.20101 42233 5	.00010 20367 0	.00000 00416 5
.253	.20142 23351 4	.00010 22450 5	.00000 00417 5
.254	.20203 04467 2	.00010 24532 3	.00000 00420 6
0.255	0.20243 65605 1	0.00010 26614 1	0.00000 00421 6
.256	.20304 46723 0	.00010 30675 7	.00000 00422 7
.257	.20345 30040 6	.00010 32757 5	.00000 00424 0
.258	.20406 11156 5	.00010 35041 3	.00000 00425 0
.259	.20446 72274 3	.00010 37123 1	.00000 00426 1
0.260	0.20507 53412 2	0.00010 41204 7	0.00000 00427 1
.261	.20550 34530 1	.00010 43266 5	.00000 00430 2
.262	.20611 15645 7	.00010 45350 3	.00000 00431 3
.263	.20651 76763 6	.00010 47432 1	.00000 00432 3
.264	.20712 60101 4	.00010 51513 7	.00000 00433 4
0.265	0.20753 41217 3	0.00010 53575 5	0.00000 00434 4
.266	.21014 22335 1	.00010 55657 3	.00000 00435 5
.267	.21055 03453 0	.00010 57741 1	.00000 00436 5
.268	.21115 64570 7	.00010 62022 6	.00000 00437 6
.269	.21156 45706 5	.00010 64104 4	.00000 00440 7
0.270	0.21217 27024 4	0.00010 66166 2	0.00000 00441 7
.271	.21260 10142 2	.00010 70250 0	.00000 00443 0
.272	.21320 71260 1	.00010 72331 6	.00000 00444 0
.273	.21361 52376 0	.00010 74413 4	.00000 00445 1
.274	.21422 33513 6	.00010 76475 2	.00000 00446 2
0.275	0.21463 14631 5	0.00011 00557 0	0.00000 00447 2
.276	.21523 75747 3	.00011 02640 6	.00000 00450 3
.277	.21564 57065 2	.00011 04722 4	.00000 00451 3
.278	.21625 40203 1	.00011 07004 2	.00000 00452 4
.279	.21666 21320 7	.00011 11066 0	.00000 00453 5
0.280	0.21727 02436 6	0.00011 13147 6	0.00000 00454 5
.281	.21767 63554 4	.00011 15231 4	.00000 00455 6
.282	.22030 44672 3	.00011 17313 2	.00000 00456 6
.283	.22071 26010 2	.00011 21374 7	.00000 00457 7
.284	.22132 07126 0	.00011 23456 5	.00000 00461 0
0.285	0.22172 70243 7	0.00011 25540 3	0.00000 00462 0
.286	.22233 51361 5	.00011 27622 1	.00000 00463 1
.287	.22274 32477 4	.00011 31703 7	.00000 00464 1
.288	.22335 13615 2	.00011 33765 5	.00000 00465 2
.289	.22375 74733 1	.00011 36047 3	.00000 00466 2
0.290	0.22436 56051 0	0.00011 40131 1	0.00000 00467 3
.291	.22477 37166 6	.00011 42212 7	.00000 00470 4
.292	.22540 20304 5	.00011 44274 5	.00000 00471 4
.293	.22601 01422 3	.00011 46356 3	.00000 00472 5
.294	.22641 62540 2	.00011 50440 1	.00000 00473 5

$(n)_{10}$	$(n)_8$	$(n \times 10^{-3})_8$	$(n \times 10^{-6})_8$
0.295	0.22702 43656 1	0.00011 52521 7	0.00000 00474 6
.296	.22743 24773 7	.00011 54603 5	.00000 00475 7
.297	.23004 06111 6	.00011 56665 3	.00000 00476 7
.298	.23044 67227 4	.00011 60747 1	.00000 00500 0
.299	.23105 50345 .3	.00011 63030 6	.00000 00501 0
0.300	0.23146 31463 2	0.00011 65112 4	0.00000 00502 1
.301	.23207 12601 0	.00011 67174 2	.00000 00503 2
.302	.23247 73716 7	.00011 71256 0	.00000 00504 2
.303	.23310 55034 5	.00011 73337 6	.00000 00505 3
.304	.23351 36152 4	.00011 75421 4	.00000 00506 3
0.305	0.23412 17270 2	0.00011 77503 2	0.00000 00507 4
.306	.23453 00406 1	.00012 01565 0	.00000 00510 5
.307	.23513 61524 0	.00012 03646 6	.00000 00511 5
.308	.23554 42641 6	.00012 05730 4	.00000 00512 6
.309	.23615 23757 5	.00012 10012 2	.00000 00513 6
0.310	0.23656 05075 4	0.00012 12074 0	0.00000 00514 7
.311	.23716 66213 2	.00012 14155 6	.00000 00515 7
.312	.23757 47331 1	.00012 16237 4	.00000 00517 0
.313	.24020 30446 7	.00012 20321 2	.00000 00520 1
.314	.24061 11564 6	.00012 22402 7	.00000 00521 1
0.315	0.24121 72702 4	0.00012 24464 5	0.00000 00522 2
.316	.24162 54020 3	.00012 26546 3	.00000 00523 2
.317	.24223 35136 2	.00012 30630 1	.00000 00524 3
.318	.24264 16254 0	.00012 32711 7	.00000 00525 4
.319	.24324 77371 7	.00012 34773 5	.00000 00526 4
0.320	0.24365 60507 5	0.00012 37055 3	0.00000 00527 5
.321	.24426 41625 4	.00012 41137 1	.00000 00530 5
.322	.24467 22743 3	.00012 43220 7	.00000 00531 6
.323	.24530 04061 1	.00012 45302 5	.00000 00532 7
.324	.24570 65177 0	.00012 47364 3	.00000 00533 7
0.325	0.24631 46314 6	0.00012 51446 1	0.00000 00535 0
.326	.24672 27432 5	.00012 53527 7	.00000 00536 0
.327	.24733 10550 4	.00012 55611 5	.00000 00537 1
.328	.24773 71666 2	.00012 57673 3	.00000 00540 1
.329	.25034 53004 1	.00012 61755 0	.00000 00541 2
0.330	0.25075 34121 7	0.00012 64036 6	0.00000 00542 3
.331	.25136 15237 6	.00012 66120 4	.00000 00543 3
.332	.25176 76355 5	.00012 70202 2	.00000 00544 4
.333	.25237 57473 3	.00012 72264 0	.00000 00545 4
.334	.25300 40611 2	.00012 74345 6	.00000 00546 5
0.335	0.25341 21727 0	0.00012 76427 4	0.00000 00547 6
.336	.25402 03044 7	.00013 00511 2	.00000 00550 6
.337	.25442 64162 6	.00013 02573 0	.00000 00551 7
.338	.25503 45300 4	.00013 04654 6	.00000 00552 7
.339	.25544 26416 3	.00013 06736 4	.00000 00554 0
0.340	0.25605 07534 1	0.00013 11020 2	0.00000 00555 1
.341	.25645 70652 0	.00013 13102 0	.00000 00556 1
.342	.25706 51767 6	.00013 15163 6	.00000 00557 2
.343	.25747 33105 5	.00013 17245 4	.00000 00560 2
.344	.26010 14223 4	.00013 21327 1	.00000 00561 3

$(n)_{10}$	$(n)_8$	$(n \times 10^{-3})_8$	$(n \times 10^{-6})_8$
0.345	0.26050 75341 2	0.00013 23410 7	0.00000 00562 4
.346	.26111 54657 1	.00013 25472 5	.00000 00563 4
.347	.26152 37574 7	.00013 27554 3	.00000 00564 5
.348	.26213 20712 6	.00013. 31636 1	.00000 00565 5
.349	.26254 02030 5	.00013 33717 7	.00000 00566 6
0.350	0.26314 63146 3	0.00013 36001 5	0.00000 00567 6
.351	.26355 44264 2	.00013 40063 3	.00000 00570 7
.352	.26416 25402 0	.00013 42145 1	.00000 00572 0
.353	.26457 06517 7	.00013 44226 7	.00000 00573 0
.354	.26517 67635 6	.00013 46310 5	.00000 00574 1
0.355	0.26560 50753 4	0.00013 50372 3	0.00000 00575 1
.356	.26621 32071 3	.00013 52454 1	.00000 00576 2
.357	.26662 13207 1	.00013 54535 7	.00000 00577 3
.358	.26722 74325 0	.00013 56617 5	.00000 00600 3
.359	.26763 55442 7	.00013 60701 3	.00000 00601 4
0.360	0.27024 36560 5	0.00013 62763 0	0.00000 00602 4
.361	.27065 17676 4	.00013 65044 6	.00000 00603 5
.362	.27126 01014 2	.00013 67126 4	.00000 00604 6
.363	.27166 62132 1	.00013 71210 2	.00000 00605 6
.364	.27227 43247 7	.00013 73272 0	.00000 00606 7
0.365	0.27270 24365 6	0.00013 75353 6	0.00000 00607 7
.366	.27331 05503 5	.00013 77435 4	.00000 00611 0
.367	.27371 66621 3	.00014 01517 2	.00000 00612 0
.368	.27432 47737 2	.00014 03601 0	.00000 00613 1
.369	.27473 31055 0	.00014 05662 6	.00000 00614 2
0.370	0.27534 12172 7	0.00014 07744 4	0.00000 00615 2
.371	.27574 73310 6	.00014 12026 2	.00000 00616 3
.372	.27635 54426 4	.00014 14110 0	.00000 00617 3
.373	.27676 35544 3	.00014 16171 6	.00000 00620 4
.374	.27737 16662 1	.00014 20253 4	.00000 00621 5
0.375	0.3000 00000 0	0.00014 22335 1	0.00000 00622 5
.376	.30040 61115 7	.00014 24416 7	.00000 00623 6
.377	.30101 42233 5	.00014 26500 5	.00000 00624 6
.378	.30142 23351 4	.00014 30562 3	.00000 00625 7
.379	.30203 04467 2	.00014 32644 1	.00000 00627 0
0.380	0.30243 65605 1	0.00014 34725 7	0.00000 00630 0
.381	.30304 46723 0	.00014 37007 5	.00000 00631 1
.382	.30345 30040 6	.00014 41071 3	.00000 00632 1
.383	.30406 11156 5	.00014 43153 1	.00000 00633 2
.384	.30446 72274 3	.00014 45234 7	.00000 00634 3
0.385	0.30507 53412 2	0.00014 47316 5	0.00000 00635 3
.386	.30550 34530 1	.00014 51400 3	.00000 00636 4
.387	.30611 15645 7	.00014 53462 1	.00000 00637 4
.388	.30651 76763 6	.00014 55543 7	.00000 00640 5
.389	.30712 60101 4	.00014 57625 5	.00000 00641 5
0.390	0.30753 41217 3	0.00014 61707 2	0.00000 00642 6
.391	.31014 22335 1	.00014 63771 0	.00000 00643 7
.392	.31055 03453 0	.00014 66052 6	.00000 00644 7
.393	.31115 64570 7	.00014 70134 4	.00000 00646 0
.394	.31156 45706 5	.00014 72216 2	.00000 00647 0

$(n)_{10}$	$(n)_8$	$(n \times 10^{-3})_8$	$(n \times 10^{-6})_8$
0.395	0.31217 27024 4	0.00014 74300 0	0.00000 00650 1
.396	.31260 10142 2	.00014 76361 6	.00000 00651 2
.397	.31320 71260 1	.00015 00443 4	.00000 00652 2
.398	.31361 52376 0	.00015 02525 2	.00000 00653 3
.399	.31422 33513 6	.00015 04607 0	.00000 00654 3
0.400	0.31463 14631 5	0.00015 06670 6	0.00000 00655 4
.401	.31523 75747 3	.00015 10752 4	.00000 00656 5
.402	.31564 57065 2	.00015 13034 2	.00000 00657 5
.403	.31625 40203 1	.00015 15116 0	.00000 00660 6
.404	.31666 21320 7	.00015 17177 6	.00000 00661 6
0.405	0.31727 02436 6	0.00015 21261 4	0.00000 00662 7
.406	.31767 63554 4	.00015 23343 1	.00000 00663 7
.407	.32030 44672 3	.00015 25424 7	.00000 00665 0
.408	.32071 26010 2	.00015 27506 5	.00000 00666 1
.409	.32132 07126 0	.00015 31570 3	.00000 00667 1
0.410	0.32172 70243 7	0.00015 33652 1	0.00000 00670 2
.411	.32233 51361 5	.00015 35733 7	.00000 00671 2
.412	.32274 32477 4	.00015 40015 5	.00000 00672 3
.413	.32335 13615 2	.00015 42077 3	.00000 00673 4
.414	.32375 74733 1	.00015 44161 1	.00000 00674 4
0.415	0.32436 56051 0	0.00015 46242 7	0.00000 00675 5
.416	.32477 37166 6	.00015 50324 5	.00000 00676 5
.417	.32540 20304 5	.00015 52406 3	.00000 00677 6
.418	.32601 01422 3	.00015 54470 1	.00000 00700 7
.419	.32641 62540 2	.00015 56551 7	.00000 00701 7
0.420	0.32702 43656 1	0.00015 60633 5	0.00000 00703 0
.421	.32743 24773 7	.00015 62715 2	.00000 00704 0
.422	.33004 06111 6	.00015 64777 0	.00000 00705 1
.423	.33044 67227 4	.00015 67060 6	.00000 00706 2
.424	.33105 50345 3	.00015 71142 4	.00000 00707 2
0.425	0.33146 31463 2	0.00015 73224 2	0.00000 00710 3
.426	.33207 12601 0	.00015 75306 0	.00000 00711 3
.427	.33247 73716 7	.00015 77367 6	.00000 00712 4
.428	.33310 55034 5	.00016 01451 4	.00000 00713 4
.429	.33351 36152 4	.00016 03533 2	.00000 00714 5
0.430	0.33412 17270 3	0.00016 05615 0	0.00000 00715 6
.431	.33453 00406 1	.00016 07676 6	.00000 00716 6
.432	.33513 61524 0	.00016 11760 4	.00000 00717 7
.433	.33554 42641 6	.00016 14042 2	.00000 00720 7
.434	.33615 23757 5	.00016 16124 0	.00000 00722 0
0.435	0.33656 05075 4	0.00016 20205 6	0.00000 00723 1
.436	.33716 66213 2	.00016 22267 3	.00000 00724 1
.437	.33757 47331 1	.00016 24351 1	.00000 00725 2
.438	.34020 30446 7	.00016 26432 7	.00000 00726 2
.439	.34061 11564 6	.00016 30514 5	.00000 00727 3
0.440	0.34121 72702 4	0.00016 32576 3	0.00000 00730 4
.441	.34162 54020 3	.00016 34660 1	.00000 00731 4
.442	.34223 35136 2	.00016 36741 7	.00000 00732 5
.443	.34264 16254 0	.00016 41023 5	.00000 00733 5
.444	.34324 77371 7	.00016 43105 3	.00000 00734 6

$(n)_{10}$	$(n)_8$			$(n \times 10^{-3})_8$			$(n \times 10^{-6})_8$		
0.445	0.34365	60507	5	0.00016	45167	1	0.00000	00735	7
.446	.34426	41625	4	.00016	47250	7	.00000	00736	7
.447	.34467	22743	3	.00016	51332	5	.00000	00740	0
.448	.34530	04061	1	.00016	53414	3	.00000	00741	0
.449	.34570	65177	0	.00016	55476	1	.00000	00742	1
0.450	0.34631	46314	6	0.00016	57557	7	0.00000	00743	1
.451	.34672	27432	5	.00016	61641	5	.00000	00744	2
.452	.34733	10550	4	.00016	63723	2	.00000	00745	3
.453	.34773	71666	2	.00016	66005	0	.00000	00746	3
.454	.35034	53004	1	.00016	70066	6	.00000	00747	4
0.455	0.35075	34121	7	0.00016	72150	4	0.00000	00750	4
.456	.35136	15237	6	.00016	74232	2	.00000	00751	5
.457	.35176	76355	5	.00016	76314	0	.00000	00752	6
.458	.35237	57473	3	.00017	00375	6	.00000	00753	6
.459	.35300	40611	2	.00017	02457	4	.00000	00754	7
0.460	0.35341	21727	0	0.00017	04541	2	0.00000	00755	7
.461	.35402	03044	7	.00017	06623	0	.00000	00757	0
.462	.35442	64162	6	.00017	10704	6	.00000	00760	1
.463	.35503	45300	4	.00017	12766	4	.00000	00761	1
.464	.35544	26416	3	.00017	15050	2	.00000	00762	2
0.465	0.35605	07534	1	0.00017	17132	0	0.00000	00763	2
.466	.35645	70652	0	.00017	21213	6	.00000	00764	3
.467	.35706	51767	6	.00017	23275	3	.00000	00765	3
.468	.35747	33105	5	.00017	25357	1	.00000	00766	4
.469	.36010	14223	4	.00017	27440	7	.00000	00767	5
0.470	0.36050	75341	2	0.00017	31522	5	0.00000	00770	5
.471	.36111	56457	1	.00017	33604	3	.00000	00771	6
.472	.36152	37574	7	.00017	35666	1	.00000	00772	6
.473	.36213	20712	6	.00017	37747	7	.00000	00773	7
.474	.36254	02030	5	.00017	42031	5	.00000	00775	0
0.475	0.36314	63146	3	0.00017	44113	3	0.00000	00776	0
.476	.36355	44264	2	.00017	46175	1	.00000	00777	1
.477	.36416	25402	0	.00017	50256	7	.00000	01000	1
.478	.36457	06517	7	.00017	52340	5	.00000	01001	2
.479	.36517	67635	6	.00017	54422	3	.00000	01002	3
0.480	0.36560	50753	4	0.00017	56504	1	0.00000	01003	3
.481	.36621	32071	3	.00017	60565	7	.00000	01004	4
.482	.36662	13207	1	.00017	62647	4	.00000	01005	4
.483	.36722	74325	0	.00017	64731	2	.00000	01006	5
.484	.36763	55442	7	.00017	67013	0	.00000	01007	6
0.485	0.37024	36560	5	0.00017	71074	6	0.00000	01010	6
.486	.37065	17676	4	.00017	73156	4	.00000	01011	7
.487	.37126	01014	2	.00017	75240	2	.00000	01012	7
.488	.37166	62132	1	.00017	77322	0	.00000	01014	0
.489	.37227	43247	7	.00020	01403	6	.00000	01015	0
0.490	0.37270	24365	6	0.00020	03465	4	0.00000	01016	1
.491	.37331	05503	5	.00020	05547	2	.00000	01017	2
.492	.37371	66621	3	.00020	07631	0	.00000	01020	2
.493	.37432	47737	2	.00020	11712	6	.00000	01021	3
.494	.37473	31055	0	.00020	13774	4	.00000	01022	3

$(n)_{10}$	$(n)_8$	$(n \times 10^{-3})_8$	$(n \times 10^{-6})_8$
0.495	0.37534 12172 7	0.00020 16056 2	0.00000 01023 4
.496	.37574 73310 6	.00020 20140 0	.00000 01024 5
.497	.37635 54426 4	.00020 22221 5	.00000 01025 5
.498	.37676 35544 3	.00020 24303 3	.00000 01026 6
.499	.37737 16662 1	.00020 26365 1	.00000 01027 6
0.500	0.40000 00000 0	0.00020 30446 7	0.00000 01030 7
.501	.40040 61115 7	.00020 32530 5	.00000 01032 0
.502	.40101 42233 5	.00020 34612 3	.00000 01033 0
.503	.40142 23351 4	.00020 36674 1	.00000 01034 1
.504	.40203 04467 2	.00020 40755 7	.00000 01035 1
0.505	0.40243 65605 1	0.00020 43037 5	0.00000 01036 2
.506	.40304 46723 0	.00020 45121 3	.00000 01037 2
.507	.40345 30040 6	.00020 47203 1	.00000 01040 3
.508	.40406 11156 5	.00020 51264 7	.00000 01041 4
.509	.40446 72274 3	.00020 53346 5	.00000 01042 4
0.510	0.40507 53412 2	0.00020 55430 3	0.00000 01043 5
.511	.40550 34530 1	.00020 57512 1	.00000 01044 5
.512	.40611 15645 7	.00020 61573 7	.00000 01045 6
.513	.40651 76763 6	.00020 63655 4	.00000 01046 7
.514	.40712 60101 4	.00020 65737 2	.00000 01047 7
0.515	0.40753 41217 3	0.00020 70021 0	0.00000 01051 0
.516	.41014 22335 1	.00020 72102 6	.00000 01052 0
.517	.41055 03453 0	.00020 74164 4	.00000 01053 1
.518	.41115 64570 7	.00020 76246 2	.00000 01054 2
.519	.41156 45706 5	.00021 00330 0	.00000 01055 2
0.520	0.41217 27024 4	0.00021 02411 6	0.00000 01056 3
.521	.41260 10142 2	.00021 04473 4	.00000 01057 3
.522	.41320 71260 1	.00021 06555 2	.00000 01060 4
.523	.41361 52376 0	.00021 10637 0	.00000 01061 5
.524	.41422 33513 6	.00021 12720 6	.00000 01062 5
0.525	0.41463 14631 5	0.00021 15002 4	0.00000 01063 6
.526	.41523 75747 3	.00021 17064 2	.00000 01064 6
.527	.41564 57065 2	.00021 21146 0	.00000 01065 7
.528	.41625 40203 1	.00021 23227 5	.00000 01066 7
.529	.41666 21320 7	.00021 25311 3	.00000 01070 0
0.530	0.41727 02436 6	0.00021 27373 1	0.00000 01071 1
.531	.41767 63554 4	.00021 31454 7	.00000 01072 1
.532	.42030 44672 3	.00021 33536 5	.00000 01073 2
.533	.42071 26010 2	.00021 35620 3	.00000 01074 2
.534	.42132 07126 0	.00021 37702 1	.00000 01075 3
0.535	0.42172 70243 7	0.00021 41763 7	0.00000 01076 4
.536	.42233 51361 5	.00021 44045 5	.00000 01077 4
.537	.42274 32477 4	.00021 46127 3	.00000 01100 5
.538	.42335 13615 2	.00021 50211 1	.00000 01101 5
.539	.42375 74733 1	.00021 52272 7	.00000 01102 6
0.540	0.42436 56051 0	0.00021 54354 5	0.00000 01103 7
.541	.42477 37166 6	.00021 56436 3	.00000 01104 7
.542	.42540 20304 5	.00021 60520 1	.00000 01106 0
.543	.42601 01422 3	.00021 62601 6	.00000 01107 0
.544	.42641 62540 2	.00021 64663 4	.00000 01110 1

$(n)_{10}$	$(n)_8$			$(n \times 10^{-3})_8$			$(n \times 10^{-6})_8$		
0.545	0.42702	43656	1	0.00021	66745	2	0.00000	01111	1
.546	.42743	24773	7	.00021	71027	0	.00000	01112	2
.547	.43004	06111	6	.00021	73110	6	.00000	01113	3
.548	.43044	67227	4	.00021	75172	4	.00000	01114	3
.549	.43105	50345	3	.00021	77254	2	.00000	01115	4
0.550	0.43146	31463	2	0.00022	01336	0	0.00000	01116	4
.551	.43207	12601	0	.00022	03417	6	.00000	01117	5
.552	.43247	73716	7	.00022	05501	4	.00000	01120	6
.553	.43310	55034	5	.00022	07563	2	.00000	01121	6
.554	.43351	36152	4	.00022	11645	0	.00000	01122	7
0.555	0.43412	17270	3	0.00022	13726	6	0.00000	01123	7
.556	.43453	00406	1	.00022	16010	4	.00000	01125	0
.557	.43513	61524	0	.00022	20072	2	.00000	01126	1
.558	.43554	42641	6	.00022	22154	0	.00000	01127	1
.559	.43615	23757	5	.00022	24235	5	.00000	01130	2
0.560	0.43656	05075	4	0.00022	26317	3	0.00000	01131	2
.561	.43716	66213	2	.00022	30401	1	.00000	01132	3
.562	.43757	47331	1	.00022	32462	7	.00000	01133	4
.563	.44020	30446	7	.00022	34544	5	.00000	01134	4
.564	.44061	11564	6	.00022	36626	3	.00000	01135	5
0.565	0.44121	72702	4	0.00022	40710	1	0.00000	01136	5
.566	.44162	54020	3	.00022	42771	7	.00000	01137	6
.567	.44223	35136	2	.00022	45053	5	.00000	01140	6
.568	.44264	16254	0	.00022	47135	3	.00000	01141	7
.569	.44324	77371	7	.00022	51217	1	.00000	01143	0
0.570	0.44365	60507	5	0.00022	53300	7	0.00000	01144	0
.571	.44426	41625	4	.00022	55362	5	.00000	01145	1
.572	.44467	22743	3	.00022	57444	3	.00000	01146	1
.573	.44530	04061	1	.00022	61526	1	.00000	01147	2
.574	.44570	65177	0	.00022	63607	6	.00000	01150	3
0.575	0.44631	46314	6	0.00022	65671	4	0.00000	01151	3
.576	.44672	27432	5	.00022	67753	2	.00000	01152	4
.577	.44733	10550	4	.00022	72035	0	.00000	01153	4
.578	.44773	71666	2	.00022	74116	6	.00000	01154	5
.579	.45034	53004	1	.00022	76200	4	.00000	01155	6
0.580	0.45075	34121	7	0.00023	00262	2	0.00000	01156	6
.581	.45135	15237	6	.00023	02344	0	.00000	01157	7
.582	.45176	76355	5	.00023	04425	6	.00000	01160	7
.583	.45237	57473	3	.00023	06507	4	.00000	01162	0
.584	.45300	40611	2	.00023	10571	2	.00000	01163	1
0.585	0.45341	21727	0	0.00023	12653	0	0.00000	01164	1
.586	.45402	03044	7	.00023	14734	6	.00000	01165	2
.587	.45442	64162	6	.00023	17016	4	.00000	01166	2
.588	.45503	45300	4	.00023	21100	2	.00000	01167	3
.589	.45544	26416	3	.00023	23161	7	.00000	01170	3
0.590	0.45605	07534	1	0.00023	25243	5	0.00000	01171	4
.591	.45645	70642	0	.00023	27325	3	.00000	01172	5
.592	.45706	51767	6	.00023	31407	1	.00000	01173	5
.593	.45747	33105	5	.00023	33470	7	.00000	01174	6
.594	.46010	14223	4	.00023	35552	5	.00000	01175	6

$(n)_{10}$	$(n)_8$	$(n \times 10^{-3})_8$	$(n \times 10^{-6})_8$
0.595	0.46050 75341 2	0.00023 37634 3	0.00000 01176 7
.596	.46111 56457 1	.00023 41716 1	.00000 01200 0
.597	.46152 37574 7	.00023 43777 7	.00000 01201 0
.598	.46213 20712 6	.00023 46061 5	.00000 01202 1
.599	.46254 02030 5	.00023 50143 3	.00000 01203 1
0.600	0.46314 63146 3	0.00023 52225 1	0.00000 01204 2
.601	.46355 44264 2	.00023 54306 7	.00000 01205 3
.602	.46416 25402 0	.00023 56370 5	.00000 01206 3
.603	.46457 06517 7	.00023 60452 3	.00000 01207 4
.604	.46517 67635 6	.00023 62534 0	.00000 01210 4
0.605	0.46560 50753 4	0.00023 64615 6	0.00000 01211 5
.606	.46621 32071 3	.00023 66677 4	.00000 01212 5
.607	.46662 13207 1	.00023 70761 2	.00000 01213 6
.608	.46722 74325 0	.00023 73043 0	.00000 01214 7
.609	.46763 55442 7	.00023 75124 6	.00000 01215 7
0.610	0.47024 36560 5	0.00023 77206 4	0.00000 01217 0
.611	.47065 17676 4	.00024 01270 2	.00000 01220 0
.612	.47126 01014 2	.00024 03352 0	.00000 01221 1
.613	.47166 62132 1	.00024 05433 6	.00000 01222 2
.614	.47227 43247 7	.00024 07515 4	.00000 01223 2
0.615	0.47270 24365 6	0.00024 11577 2	0.00000 01224 3
.616	.47331 05503 5	.00024 13661 0	.00000 01225 3
.617	.47371 66621 3	.00024 15742 6	.00000 01226 4
.618	.47432 47737 2	.00024 20024 4	.00000 01227 5
.619	.47473 31055 0	.00024 22106 2	.00000 01230 5
0.620	0.47534 12172 7	0.00024 24167 7	0.00000 01231 6
.621	.47574 73310 6	.00024 26251 5	.00000 01232 6
.622	.47635 54426 4	.00024 30333 3	.00000 01233 7
.623	.47676 35544 3	.00024 32415 1	.00000 01235 0
.624	.47737 16662 1	.00024 34476 7	.00000 01236 0
0.625	0.50000 00000 0	0.00024 36560 5	0.00000 01237 1
.626	.50040 61115 7	.00024 40642 3	.00000 01240 1
.627	.50101 42233 5	.00024 42724 1	.00000 01241 2
.628	.50142 23351 4	.00024 45005 7	.00000 01242 2
.629	.50203 04467 2	.00024 47067 5	.00000 01243 3
0.630	0.50243 65605 1	0.00024 51151 3	0.00000 01244 4
.631	.50304 46723 0	.00024 53233 1	.00000 01245 4
.632	.50345 30040 6	.00024 55314 7	.00000 01246 5
.633	.50406 11156 5	.00024 57376 5	.00000 01247 5
.634	.50446 72274 3	.00024 61460 3	.00000 01250 6
0.635	0.50507 53412 2	0.00024 63542 0	0.00000 01251 7
.636	.50550 34530 1	.00024 65623 6	.00000 01252 7
.637	.50611 15645 7	.00024 67705 4	.00000 01254 0
.638	.50651 76763 6	.00024 71767 2	.00000 01255 0
.639	.50712 60101 4	.00024 74051 0	.00000 01256 1
0.640	0.50753 41217 3	0.00024 76132 6	0.00000 01257 2
.641	.51014 22335 1	.00025 00214 4	.00000 01260 2
.642	.51055 03453 0	.00025 02276 2	.00000 01261 3
.643	.51115 64570 7	.00025 04360 0	.00000 01262 3
.644	.51156 45706 5	.00025 06441 6	.00000 01263 4

$(n)_{10}$	$(n)_8$	$(n \times 10^{-3})_8$	$(n \times 10^{-6})_8$
0.645	0.51217 27024 4	0.00025 10523 4	0.00000 01264 4
.646	.51260 10142 2	.00025 12605 2	.00000 01265 5
.647	.51320 71260 1	.00025 14667 0	.00000 01266 6
.648	.51361 52376 0	.00025 16750 6	.00000 01267 6
.649	.51422 33513 6	.00025 21032 4	.00000 01270 7
0.650	0.51463 14631 5	0.00025 23114 1	0.00000 01271 7
.651	.51523 75747 3	.00025 25175 7	.00000 01273 0
.652	.51564 57065 2	.00025 27257 5	.00000 01274 1
.653	.51625 40203 1	.00025 31341 3	.00000 01275 1
.654	.51666 21320 7	.00025 33423 1	.00000 01276 2
0.655	0.51727 02436 6	0.00025 35504 7	0.00000 01277 2
.656	.51767 63554 4	.00025 37566 5	.00000 01300 3
.657	.52030 44672 3	.00025 41650 3	.00000 01301 4
658	.52071 26010 2	.00025 43732 1	.00000 01302 4
.659	.52132 07126 0	.00025 46013 7	.00000 01303 5
0.660	0.52172 70243 7	0.00025 50075 5	0.00000 01304 5
.661	.52233 51361 5	.00025 52157 3	.00000 01305 6
.662	.52274 32477 4	.00025 54241 1	.00000 01305 7
.663	.52335 13615 2	.00025 56322 7	.00000 01307 7
.664	.52375 74733 1	.00025 60404 5	.00000 01311 0
0.665	0.52436 56051 0	0.00025 62466 3	0.00000 01312 0
.666	.52477 37166 6	.00025 64550 0	.00000 01313 1
.667	52540 20304 5	.00025 66631 6	.00000 01314 1
.668	.52601 01422 3	.00025 70713 4	.00000 01315 2
.669	.52641 62540 2	.00025 72775 2	.00000 01316 3
0.670	0.52702 43656 1	0.00025 75057 0	0.00000 01317 3
.671	.52743 24773 7	.00025 77140 6	.00000 01320 4
.672	.53004 06111 6	.00026 01222 4	.00000 01321 4
.673	.53044 67227 4	.00026 03304 2	.00000 01322 5
.674	.53105 50345 3	.00026 05366 0	.00000 01323 6
0.675	0.53146 31463 2	0.00026 07447 6	0.00000 01324 6
.676	.53207 12601 0	.00026 11531 4	.00000 01325 7
.677	.53247 73716 7	.00026 13613 2	.00000 01326 7
.678	.53310 55034 5	.00026 15675 0	.00000 01330 0
.679	.53351 36152 4	.00026 17756 6	.00000 01331 1
0.680	0.53412 17270 3	0.00026 22040 4	0.00000 01332 1
.681	.53453 00406 1	.00026 24122 1	.00000 01333 2
.682	.53513 61524 0	.00026 26203 7	.00000 01334 2
.683	.53554 42641 6	.00026 30265 5	.00000 01335 3
.684	.53615 23757 5	.00026 32347 3	.00000 01336 3
0.685	0.53656 05075 4	0.00026 34431 1	0.00000 01337 4
.686	.53716 66213 2	.00026 36512 7	.00000 01340 5
.687	.53757 47331 1	.00026 40574 5	.00000 01341 5
.688	.54020 30446 7	.00026 42656 3	.00000 01342 6
.689	.54061 11564 6	.00026 44740 1	.00000 01343 6
0.690	0.54121 72702 4	0.00026 47021 7	0.00000 01344 7
.691	.54162 54020 3	.00026 51103 5	.00000 01346 0
.692	.54223 35136 2	.00026 53165 3	.00000 01347 0
.693	.54264 16254 0	.00026 55247 1	.00000 01350 1
.694	.54324 77371 7	.00026 57330 7	.00000 01351 1

$(n)_{10}$	$(n)_8$	$(n \times 10^{-3})_8$	$(n \times 10^{-6})_8$
0.695	0.54365 60507 5	0.00026 61412 5	0.00000 01352 2
.696	.54426 41625 4	.00026 63474 2	.00000 01353 3
.697	.54467 22743 3	.00026 65556 0	.00000 01354 3
.698	.54530 04061 1	.00026 67637 6	.00000 01355 4
.699	.54570 65177 0	.00026 71721 4	.00000 01356 4
0.700	0.54631 46314 6	0.00026 74003 2	0.00000 01357 5
.701	.54672 27432 5	.00026 76065 0	.00000 01360 6
.702	.54733 10550 4	.00027 00146 6	.00000 01361 6
.703	.54773 71666 2	.00027 02230 4	.00000 01362 7
.704	.55034 53004 1	.00027 04312 2	.00000 01363 7
0.705	0.55075 34121 7	0.00027 06374 0	0.00000 01365 0
.706	.55136 15237 6	.00027 10455 6	.00000 01366 0
.707	.55176 76355 5	.00027 12537 4	.00000 01367 1
.708	.55237 57473 3	.00027 14621 2	.00000 01370 2
.709	.55300 40611 2	.00027 16703 0	.00000 01371 2
0.710	0.55341 21727 0	0.00027 20764 6	0.00000 01372 3
.711	.55402 03044 7	.00027 23046 3	.00000 01373 3
.712	.55442 64162 6	.00027 25130 1	.00000 01374 4
.713	.55503 45300 4	.00027 27211 7	.00000 01375 5
.714	.55544 26416 3	.00027 31273 5	.00000 01376 5
0.715	0.55605 07534 1	0.00027 33355 3	0.00000 01377 6
.716	.55645 70652 0	.00027 35437 1	.00000 01400 6
.717	.55706 51767 6	.00027 37520 7	.00000 01401 7
.718	.55747 33105 5	.00027 41602 5	.00000 01403 0
.719	.56010 14223 4	.00027 43664 3	.00000 01404 0
0.720	0.56050 75341 2	0.00027 45746 1	0.00000 01405 1
.721	.56111 56457 1	.00027 50027 7	.00000 01406 1
.722	.56152 37574 7	.00027 52111 5	.00000 01407 2
.723	.56213 20712 6	.00027 54173 3	.00000 01410 3
.724	.56254 02030 5	.00027 56255 1	.00000 01411 3
0.725	0.56314 63146 3	0.00027 60336 7	0.00000 01412 4
.726	.56355 44264 2	.00027 62420 5	.00000 01413 4
.727	.56416 25402 0	.00027 64502 2	.00000 01414 5
.728	.56457 06517 7	.00027 66564 0	.00000 01415 5
.729	.56517 67635 6	.00027 70645 6	.00000 01416 6
0.730	0.56560 50753 4	0.00027 72727 4	0.00000 01417 7
.731	.56621 32071 3	.00027 75011 2	.00000 01420 7
.732	.56662 13207 1	.00027 77073 0	.00000 01422 0
.733	.56722 74325 0	.00030 01154 6	.00000 01423 0
.734	.56763 55442 7	.00030 03236 4	.00000 01424 1
0.735	0.57024 36560 5	0.00030 05320 2	0.00000 01425 2
.736	.57065 17676 4	.00030 07402 0	.00000 01426 2
.737	.57126 01014 2	.00030 11463 6	.00000 01427 3
.738	.57166 62132 1	.00030 13545 4	.00000 01430 3
.739	.57227 43247 7	.00030 15627 2	.00000 01431 4
0.740	0.57270 24365 6	0.00030 17711 0	0.00000 01432 5
.741	.57331 05503 5	.00030 21772 6	.00000 01433 5
.742	.57371 66621 3	.00030 24054 3	.00000 01434 6
.743	.57432 47737 2	.00030 26136 1	.00000 01435 6
.744	.57473 31055 0	.00030 30217 7	.00000 01436 7

$(n)_{10}$	$(n)_8$	$(n \times 10^{-3})_8$	$(n \times 10^{-6})_8$
0.745	0.57534 12172 7	0.00030 32301 5	0.00000 01437 7
.746	.57574 73310 6	.00030 34363 3	.00000 01441 0
.747	.57635 54426 4	.00030 36445 1	.00000 01442 1
.748	.57676 35544 3	.00030 40526 7	.00000 01443 1
.749	.57737 16662 1	.00030 42610 5	.00000 01444 2
0.750	0.60000 00000 0	0.00030 44672 3	0.00000 01445 2
.751	.60040 61115 7	.00030 46754 1	.00000 01446 3
.752	.60101 42233 5	.00030 51035 7	.00000 01447 4
.753	.60142 23351 4	.00030 53117 5	.00000 01450 4
.754	.60203 04467 2	.00030 55201 3	.00000 01451 5
0.755	0.60243 65605 1	0.00030 57263 1	0.00000 01452 5
.756	.60304 46723 0	.00030 61344 7	.00000 01453 6
.757	.60345 30040 6	.00030 63426 4	.00000 01454 7
.758	.60406 11156 5	.00030 65510 2	.00000 01455 7
.759	.60446 72274 3	.00030 67572 0	.00000 01457 0
0.760	0.60507 53412 2	0.00030 71653 6	0.00000 01460 0
.761	.60550 34530 1	.00030 73735 4	.00000 01461 1
.762	.60611 15645 7	.00030 76017 2	.00000 01462 2
.763	.60651 76763 6	.00031 00101 0	.00000 01463 2
.764	.60712 60101 4	.00031 02162 6	.00000 01464 3
0.765	0.60753 41217 3	0.00031 04244 4	0.00000 01465 3
.766	.61014 22335 1	.00031 06326 2	.00000 01466 4
.767	.61055 03453 0	.00031 10410 0	.00000 01467 4
.768	.61115 64570 7	.00031 12471 6	.00000 01470 5
.769	.61156 45706 5	.00031 14553 4	.00000 01471 6
0.770	0.61217 27024 4	0.00031 16635 2	0.00000 01472 6
.771	.61260 10142 2	.00031 20717 0	.00000 01473 7
.772	.61320 71260 1	.00031 23000 6	.00000 01474 7
.773	.61361 52376 0	.00031 25062 3	.00000 01476 0
.774	.61422 33513 6	.00031 27144 1	.00000 01477 1
0.775	0.61463 14631 5	0.00031 31225 7	0.00000 01500 1
.776	.61523 75747 3	.00031 33307 5	.00000 01501 2
.777	.61564 57065 2	.00031 35371 3	.00000 01502 2
.778	.61625 40203 1	.00031 37453 1	.00000 01503 3
.779	.61666 21320 7	.00031 41534 7	.00000 01504 4
0.780	0.61727 02436 6	0.00031 43616 5	0.00000 01505 4
.781	.61767 63554 4	.00031 45700 3	.00000 01506 5
.782	.62030 44672 3	.00031 47762 1	.00000 01507 5
.783	.62071 26010 2	.00031 52043 7	.00000 01510 6
.784	.62132 07126 0	.00031 54125 5	.00000 01511 6
0.785	0.62172 70243 7	0.00031 56207 3	0.00000 01512 7
.786	.62233 51361 5	.00031 60271 1	.00000 01514 0
.787	.62274 32477 4	.00031 62352 7	.00000 01515 0
.788	.62335 13615 2	.00031 64434 4	.00000 01516 1
.789	.62375 74733 1	.00031 66516 2	.00000 01517 1
0.790	0.62436 56051 0	0.00031 70600 0	0.00000 01520 2
.791	.62477 37166 6	.00031 72661 6	.00000 01521 3
.792	.62540 20304 5	.00031 74743 4	.00000 01522 3
.793	.62601 01422 3	.00031 77025 2	.00000 01523 4
.794	.62641 62540 2	.00032 01107 0	.00000 01524 4

$(n)_{10}$	$(n)_8$	$(n \times 10^{-3})_8$	$(n \times 10^{-6})_8$
0.795	0.62702 43656 1	0.00032 03170 6	0.00000 01525 5
.796	.62743 24773 7	.00032 05252 4	.00000 01526 6
.797	.63004 06111 6	.00032 07334 2	.00000 01527 6
.798	.63044 67227 4	.00032 11416 0	.00000 01530 7
.799	.63105 50345 3	.00032 13477 6	.00000 01531 7
0.800	0.63146 31463 2	0.00032 15561 4	0.00000 01533 0
.801	.63207 12601 0	.00032 17643 2	.00000 01534 1
.802	.63247 73716 7	.00032 21725 0	.00000 01535 1
.803	.63310 55034 5	.00032 24006 5	.00000 01536 2
.804	.63351 36152 4	.00032 26070 3	.00000 01537 2
0.805	0.63412 17270 3	0.00032 30152 1	0.00000 01540 3
.806	.63453 00406 1	.00032 32233 7	.00000 01541 3
.807	.63513 61524 0	.00032 34315 5	.00000 01542 4
.808	.63554 42641 6	.00032 36377 3	.00000 01543 5
.809	.63615 23757 5	.00032 40461 1	.00000 01544 5
0.810	0.63656 05075 4	0.00032 42542 7	0.00000 01545 6
.811	.63716 66213 2	.00032 44624 5	.00000 01546 6
.812	.63757 47331 1	.00032 46706 3	.00000 01547 7
.813	.64020 30446 7	.00032 50770 1	.00000 01551 0
.814	.64061 11564 6	.00032 53051 7	.00000 01552 0
0.815	0.64121 72702 4	0.00032 55133 5	0.00000 01553 1
.816	.64162 54020 3	.00032 57215 3	.00000 01554 1
.817	.64223 35136 2	.00032 61277 1	.00000 01555 2
.818	.64264 16254 0	.00032 63360 6	.00000 01556 3
.819	.64324 77371 7	.00032 65442 4	.00000 01557 3
0.820	0.64365 60507 5	0.00032 67524 2	0.00000 01560 4
.821	.64426 41625 4	.00032 71606 0	.00000 01561 4
.822	.64467 22743 3	.00032 73667 6	.00000 01562 5
.823	.64530 04061 1	.00032 75751 4	.00000 01563 5
.824	.64570 65177 0	.00033 00033 2	.00000 01564 6
0.825	0.64631 46314 6	0.00033 02115 0	0.00000 01566 7
.826	.64672 27432 5	.00033 04176 6	.00000 01566 7
.827	.64733 10550 4	.00033 06260 4	.00000 01570 0
.828	.64773 71666 2	.00033 10342 2	.00000 01571 0
.829	.65034 53004 1	.00033 12424 0	.00000 01572 1
0.830	0.65075 34121 7	0.00033 14505 6	0.00000 01573 2
.831	.65136 15237 6	.00033 16567 4	.00000 01574 2
.832	.65176 76355 5	.00033 20651 2	.00000 01575 3
.833	.65237 57473 3	.00033 22733 0	.00000 01576 3
.834	.65300 40611 2	.00033 25014 5	.00000 01577 4
0.835	0.65341 21727 0	0.00033 27076 3	0.00000 01600 5
.836	.65402 03044 7	.00033 31160 1	.00000 01601 5
.837	.65442 64162 6	.00033 33241 7	.00000 01602 6
.838	.65503 45300 4	.00033 35323 5	.00000 01603 6
.839	.65544 26416 3	.00033 37405 3	.00000 01604 7
0.840	0.65605 07534 1	0.00033 41467 1	0.00000 01606 0
.841	.65645 70652 0	.00033 43550 7	.00000 01607 0
.842	.65706 51767 6	.00033 45632 5	.00000 01610 1
.843	.65747 33105 5	.00033 47714 3	.00000 01611 1
.844	.66010 14223 4	.00033 51776 1	.00000 01612 2

$(n)_{10}$	$(n)_8$	$(n \times 10^{-3})_8$	$(n \times 10^{-6})_8$
0.845	0.66050 75341 2	0.00033 54057 7	0.00000 01613 2
.846	.66111 56457 1	.00033 56141 5	.00000 01614 3
.847	.66152 37574 7	.00033 60223 3	.00000 01615 4
.848	.66213 20712 6	.00033 62305 1	.00000 01616 4
.849	.66254 02030 5	.00033 64366 6	.00000 01617 5
0.850	0.66314 63146 3	0.00033 66540 4	0.00000 01620 5
.851	.66355 44264 2	.00033 70532 2	.00000 01621 6
.852	.66416 25402 0	.00033 72614 0	.00000 01622 7
.853	.66457 06517 7	.00033 74675 6	.00000 01623 7
.854	.66517 67635 6	.00033 76757 4	.00000 01625 0
0.855	0.66560 50753 4	0.00034 01041 2	0.00000 01626 0
.856	.66621 32071 3	.00034 03123 0	.00000 01627 1
.857	.66662 13207 1	.00034 05204 6	.00000 01630 2
.858	.66722 74325 0	.00034 07266 4	.00000 01631 2
.859	.66763 55442 7	.00034 11350 2	.00000 01632 3
0.860	0.67024 36560 5	0.00034 13432 0	0.00000 01633 3
.861	.67065 17676 4	.00034 15513 6	.00000 01634 4
.862	.67126 01014 2	.00034 17575 4	.00000 01635 5
.863	.67166 62132 1	.00034 21657 2	.00000 01636 5
.864	.67227 43247 7	.00034 23740 7	.00000 01637 6
0.865	0.67270 24365 6	0.00034 26022 5	0.00000 01640 6
.866	.67331 05503 5	.00034 30104 3	.00000 01641 7
.867	.67371 66621 3	.00034 32166 1	.00000 01642 7
.868	.67432 47737 2	.00034 34247 7	.00000 01644 0
.869	.67473 31055 0	.00034 36331 5	.00000 01645 1
0.870	0.67534 12172 7	0.00034 40413 3	0.00000 01646 1
.871	.67574 73310 6	.00034 42475 1	.00000 01647 2
.872	.67635 54426 4	.00034 44556 7	.00000 01650 2
.873	.67676 35544 3	.00034 46640 5	.00000 01651 3
.874	.67737 16662 1	.00034 50722 3	.00000 01652 4
0.875	0.70000 00000 0	0.00034 53004 1	0.00000 01653 4
.876	.70040 61115 7	.00034 55065 7	.00000 01654 5
.877	.70101 42233 5	.00034 57147 5	.00000 01655 5
.878	.70142 23351 4	.00034 61231 3	.00000 01656 6
.879	.70203 04467 2	.00034 63313 1	.00000 01657 7
0.880	0.70243 65605 1	0.00034 65374 6	0.00000 01660 7
.881	.70304 46723 0	.00034 67456 4	.00000 01662 0
.882	.70345 30040 6	.00034 71540 2	.00000 01663 0
.883	.70406 11156 5	.00034 73622 0	.00000 01664 1
.884	.70446 72274 3	.00034 75703 6	.00000 01665 1
0.885	0.70507 53412 2	0.00034 77765 4	0.00000 01666 2
.886	.70550 34530 1	.00035 02047 2	.00000 01667 3
.887	.70611 15645 7	.00035 04131 0	.00000 01670 3
.888	.70651 76763 6	.00035 06212 6	.00000 01671 4
.889	.70712 60101 4	.00035 10274 4	.00000 01672 4
0.890	0.70753 41217 3	0.00035 12356 2	0.00000 01673 5
.891	.71014 22335 1	.00035 14440 0	.00000 01674 6
.892	.71055 03453 0	.00035 16521 6	.00000 01675 6
.893	.71115 64570 7	.00035 20603 4	.00000 01676 7
.894	.71156 45706 5	.00035 22665 2	.00000 01677 7

$(n)_{10}$	$(n)_8$	$(n \times 10^{-3})_8$	$(n \times 10^{-6})_8$
0.895	0.71217 27024 4	0.00035 24746 7	0.00000 01701 0
.896	.71260 10142 2	.00035 27030 5	.00000 01702 1
.897	.71320 71260 1	.00035 31112 3	.00000 01703 1
.898	.71361 52376 0	.00035 33174 1	.00000 01704 2
.899	.71422 33513 6	.00035 35255 7	.00000 01705 2
0.900	0.71463 14631 5	0.00035 37337 5	0.00000 01706 3
.901	.71523 75747 3	.00035 41421 3	.00000 01707 4
.902	.71564 57065 2	.00035 43503 1	.00000 01710 4
.903	.71625 40203 1	.00035 45564 7	.00000 01711 5
.904	.71666 21320 7	.00035 47646 5	.00000 01712 5
0.905	0.71727 02436 6	0.00035 51730 3	0.00000 01713 6
.906	.71767 63554 4	.00035 54012 1	.00000 01714 6
.907	.72030 44672 3	.00035 56073 7	.00000 01715 7
.908	.72071 26010 2	.00035 60155 5	.00000 01717 0
.909	.72132 07126 0	.00035 62237 3	.00000 01720 0
0.910	0.72172 70243 7	0.00035 64321 0	0.00000 01721 1
.911	.72233 51361 5	.00035 66402 6	.00000 01722 1
.912	.72274 32477 4	.00035 70464 4	.00000 01723 2
.913	.72335 13615 2	.00035 72546 2	.00000 01724 3
.914	.72375 74733 1	.00035 74630 0	.00000 01725 3
0.915	0.72436 56051 0	0.00035 76711 6	0.00000 01726 4
.916	.72477 37166 6	.00036 00773 4	.00000 01727 4
.917	.72540 20304 5	.00036 03055 2	.00000 01730 5
.918	.72601 01422 3	.00036 05137 0	.00000 01731 6
.919	.72641 62540 2	.00036 07220 6	.00000 01732 6
0.920	0.72702 43656 1	0.00036 11302 4	0.00000 01733 7
.921	.72743 24773 7	.00036 13364 2	.00000 01734 7
.922	.73004 06111 6	.00036 15446 0	.00000 01736 0
.923	.73044 67227 4	.00036 17527 6	.00000 01737 0
.924	.73105 50345 3	.00036 21611 4	.00000 01740 1
0.925	0.73146 31463 2	0.00036 23673 1	0.00000 01741 2
.926	.73207 12601 0	.00036 25754 7	.00000 01742 2
.927	.73247 73716 7	.00036 30036 5	.00000 01743 3
.928	.73310 55034 5	.00036 32120 3	.00000 01744 3
.929	.73351 36152 4	.00036 34202 1	.00000 01745 4
0.930	0.73412 17270 3	0.00036 36263 7	0.00000 01746 5
.931	.73453 00406 1	.00036 40345 5	.00000 01747 5
.932	.73513 61524 0	.00036 42427 3	.00000 01750 6
.933	.73554 42641 6	.00036 44511 1	.00000 01751 6
.934	.73615 23757 5	.00036 46572 7	.00000 01752 7
0.935	0.73656 05075 4	0.00036 50654 5	0.00000 01754 0
.936	.73716 66213 2	.00036 52736 3	.00000 01755 0
.937	.73757 47331 1	.00036 55020 1	.00000 01756 1
.938	.74020 30446 7	.00036 57101 7	.00000 01757 1
.939	.74061 11564 6	.00036 61163 5	.00000 01760 2
0.940	0.74121 72702 4	0.00036 63245 3	0.00000 01761 3
.941	.74162 54020 3	.00036 65327 0	.00000 01762 3
.942	.74223 35136 2	.00036 67410 6	.00000 01763 4
.943	.74264 16254 0	.00036 71472 4	.00000 01764 4
.944	.74324 77371 7	.00036 73554 2	.00000 01765 5

$(n)_{10}$	$(n)_8$			$(n \times 10^{-3})_8$			$(n \times 10^{-6})_8$		
0.945	0.74365	60507	5	0.00036	75636	0	0.00000	01766	5
.946	.74426	41625	4	.00036	77717	6	.00000	01767	6
.947	.74467	22743	3	.00037	02001	4	.00000	01770	7
.948	.74530	04061	1	.00037	04063	2	.00000	01771	7
.949	.74570	65177	0	.00037	06145	0	.00000	01773	0
0.950	0.74631	46314	6	0.00037	10226	6	0.00000	01774	0
.951	.74672	27432	5	.00037	12310	4	.00000	01775	1
.952	.74733	10550	4	.00037	14372	2	.00000	01776	2
.953	.74773	71666	2	.00037	16454	0	.00000	01777	2
.954	.75034	53004	1	.00037	20535	6	.00000	02000	3
0.955	0.75075	34121	7	0.00037	22617	4	0.00000	02001	3
.956	.75136	15237	6	.00037	24701	1	.00000	02002	4
.957	.75176	76355	5	.00037	26762	7	.00000	02003	5
.958	.75237	57473	3	.00037	31044	5	.00000	02004	5
.959	.75300	40611	2	.00037	33126	3	.00000	02005	6
0.960	0.75341	21727	0	0.00037	35210	1	0.00000	02006	6
.961	.75402	03044	7	.00037	37271	7	.00000	02007	7
.962	.75442	64162	6	.00037	41353	5	.00000	02010	7
.963	.75503	45300	4	.00037	43435	3	.00000	02012	0
.964	.75544	26416	3	.00037	45517	1	.00000	02013	1
0.965	0.75605	07534	1	0.00037	47600	7	0.00000	02014	1
.966	.75645	70652	0	.00037	51662	5	.00000	02015	2
.967	.75706	51767	6	.00037	53744	3	.00000	02016	2
.968	.75747	33105	5	.00037	56026	1	.00000	02017	3
.969	.76010	14223	4	.00037	60107	7	.00000	02020	4
0.970	0.76050	75341	2	0.00037	62171	5	0.00000	02021	4
.971	.76111	56457	1	.00037	64253	2	.00000	02022	5
.972	.76152	37574	7	.00037	66335	0	.00000	02023	5
.973	.76213	20712	6	.00037	70416	6	.00000	02024	6
.974	.76254	02030	5	.00037	72500	4	.00000	02025	7
0.975	0.76314	63146	3	0.00037	74562	2	0.00000	02026	7
.976	.76355	44264	2	.00037	76644	0	.00000	02030	0
.977	.76416	25402	0	.00040	00725	6	.00000	02031	0
.978	.76457	06517	7	.00040	03007	4	.00000	02032	1
.979	.76517	67635	6	.00040	05071	2	.00000	02033	2
0.980	0.76560	50753	4	0.00040	07153	0	0.00000	02034	2
.981	.76621	32071	3	.00040	11234	6	.00000	02035	3
.982	.76662	13207	1	.00040	13316	4	.00000	02036	3
.983	.76722	74325	0	.00040	15400	2	.00000	02037	4
.984	.76763	55442	7	.00040	17462	0	.00000	02040	4
0.985	0.77024	36560	5	0.00040	21543	6	0.00000	02041	5
.986	.77065	17676	4	.00040	23625	4	.00000	02042	6
.987	.77126	01014	2	.00040	25707	1	.00000	02043	6
.988	.77166	62132	1	.00040	27770	7	.00000	02044	7
.989	.77227	43247	7	.00040	32052	5	.00000	02045	7
0.990	0.77270	24365	6	0.00040	34134	3	0.00000	02047	0
.991	.77331	05503	5	.00040	36216	1	.00000	02050	1
.992	.77371	66621	3	.00040	40277	7	.00000	02051	1
.993	.77432	47737	2	.00040	42361	5	.00000	02052	2
.994	.77473	31055	0	.00040	44443	3	.00000	02053	2
0.995	0.77534	12172	7	0.00040	46525	1	0.00000	02054	3
.996	.77574	73310	6	.00040	50606	7	.00000	02055	4
.997	.77635	54426	4	.00040	52670	5	.00000	02056	4
.998	.77676	35544	3	.00040	54752	3	.00000	02057	5
.999	.77737	16662	1	.00040	57034	1	.00000	02060	5

APPENDIX III

Octal-Decimal Conversions

OCTAL	0	1	2	3	4	5	6	7
00000	00000	00001	00002	00003	00004	00005	00006	00007
00010	00008	00009	00010	00011	00012	00013	00014	00015
00020	00016	00017	00018	00019	00020	00021	00022	00023
00030	00024	00025	00026	00027	00028	00029	00030	00031
00040	00032	00033	00034	00035	00036	00037	00038	00039
00050	00040	00041	00042	00043	00044	00045	00046	00047
00060	00048	00049	00050	00051	00052	00053	00054	00055
00070	00056	00057	00058	00059	00060	00061	00062	00063
00100	00064	00065	00066	00067	00068	00069	00070	00071
00110	00072	00073	00074	00075	00076	00077	00078	00079
00120	00080	00081	00082	00083	00084	00085	00086	00087
00130	00088	00089	00090	00091	00092	00093	00094	00095
00140	00096	00097	00098	00099	00100	00101	00102	00103
00150	00104	00105	00106	00107	00108	00109	00110	00111
00160	00112	00113	00114	00115	00116	00117	00118	00119
00170	00120	00121	00122	00123	00124	00125	00126	00127
00200	00128	00129	00130	00131	00132	00133	00134	00135
00210	00136	00137	00138	00139	00140	00141	00142	00143
00220	00144	00145	00146	00147	00148	00149	00150	00151
00230	00152	00153	00154	00155	00156	00157	00158	00159
00240	00160	00161	00162	00163	00164	00165	00166	00167
00250	00168	00169	00170	00171	00172	00173	00174	00175
00260	00176	00177	00178	00179	00180	00181	00182	00183
00270	00184	00185	00186	00187	00188	00189	00190	00191
00300	00192	00193	00194	00195	00196	00197	00198	00199
00310	00200	00201	00202	00203	00204	00205	00206	00207
00320	00208	00209	00210	00211	00212	00213	00214	00215
00330	00216	00217	00218	00219	00220	00221	00222	00223
00340	00224	00225	00226	00227	00228	00229	00230	00231
00350	00232	00233	00234	00235	00236	00237	00238	00239
00360	00240	00241	00242	00243	00244	00245	00246	00247
00370	00248	00249	00250	00251	00252	00253	00254	00255
00400	00256	00257	00258	00259	00260	00261	00262	00263
00410	00264	00265	00266	00267	00268	00269	00270	00271
00420	00272	00273	00274	00275	00276	00277	00278	00279
00430	00280	00281	00282	00283	00284	00285	00286	00287
00440	00288	00289	00290	00291	00292	00293	00294	00295
00450	00296	00297	00298	00299	00300	00301	00302	00303
00460	00304	00305	00306	00307	00308	00309	00310	00311
00470	00312	00313	00314	00315	00316	00317	00318	00319
00500	00320	00321	00322	00323	00324	00325	00326	00327
00510	00328	00329	00330	00331	00332	00333	00334	00335
00520	00336	00337	00338	00339	00340	00341	00342	00343
00530	00344	00345	00346	00347	00348	00349	00350	00351
00540	00352	00353	00354	00355	00356	00357	00358	00359
00550	00360	00361	00362	00363	00364	00365	00366	00367
00560	00368	00369	00370	00371	00372	00373	00374	00375
00570	00376	00377	00378	00379	00380	00381	00382	00383
00600	00384	00385	00386	00387	00388	00389	00390	00391
00610	00392	00393	00394	00395	00396	00397	00398	00399
00620	00400	00401	00402	00403	00404	00405	00406	00407
00630	00408	00409	00410	00411	00412	00413	00414	00415
00640	00416	00417	00418	00419	00420	00421	00422	00423
00650	00424	00425	00426	00427	00428	00429	00430	00431
00660	00432	00433	00434	00435	00436	00437	00438	00439
00670	00440	00441	00442	00443	00444	00445	00446	00447
00700	00448	00449	00450	00451	00452	00453	00454	00455
00710	00456	00457	00458	00459	00460	00461	00462	00463
00720	00464	00465	00466	00467	00468	00469	00470	00471
00730	00472	00473	00474	00475	00476	00477	00478	00479
00740	00480	00481	00482	00483	00484	00485	00486	00487
00750	00488	00489	00490	00491	00492	00493	00494	00495
00760	00496	00497	00498	00499	00500	00501	00502	00503
00770	00504	00505	00506	00507	00508	00509	00510	00511

(Courtesy Computer Control Co.)

OCTAL	0	1	2	3	4	5	6	7
01000	00512	00513	00514	00515	00516	00517	00518	00519
01010	00520	00521	00522	00523	00524	00525	00526	00527
01020	00528	00529	00530	00531	00532	00533	00534	00535
01030	00536	00537	00538	00539	00540	00541	00542	00543
01040	00544	00545	00546	00547	00548	00549	00550	00551
01050	00552	00553	00554	00555	00556	00557	00558	00559
01060	00560	00561	00562	00563	00564	00565	00566	00567
01070	00568	00569	00570	00571	00572	00573	00574	00575
01100	00576	00577	00578	00579	00580	00581	00582	00583
01110	00584	00585	00586	00587	00588	00589	00590	00591
01120	00592	00593	00594	00595	00596	00597	00598	00599
01130	00600	00601	00602	00603	00604	00605	00606	00607
01140	00608	00609	00610	00611	00612	00613	00614	00615
01150	00616	00617	00618	00619	00620	00621	00622	00623
01160	00624	00625	00626	00627	00628	00629	00630	00631
01170	00632	00633	00634	00635	00636	00637	00638	00639
01200	00640	00641	00642	00643	00644	00645	00646	00647
01210	00648	00649	00650	00651	00652	00653	00654	00655
01220	00656	00657	00658	00659	00660	00661	00662	00663
01230	00664	00665	00666	00667	00668	00669	00670	00671
01240	00672	00673	00674	00675	00676	00677	00678	00679
01250	00680	00681	00682	00683	00684	00685	00686	00687
01260	00688	00689	00690	00691	00692	00693	00694	00695
01270	00696	00697	00698	00699	00700	00701	00702	00703
01300	00704	00705	00706	00707	00708	00709	00710	00711
01310	00712	00713	00714	00715	00716	00717	00718	00719
01320	00720	00721	00722	00723	00724	00725	00726	00727
01330	00728	00729	00730	00731	00732	00733	00734	00735
01340	00736	00737	00738	00739	00740	00741	00742	00743
01350	00744	00745	00746	00747	00748	00749	00750	00751
01360	00752	00753	00754	00755	00756	00757	00758	00759
01370	00760	00761	00762	00763	00764	00765	00766	00767
01400	00768	00769	00770	00771	00772	00773	00774	00775
01410	00776	00777	00778	00779	00780	00781	00782	00783
01420	00784	00785	00786	00787	00788	00789	00790	00791
01430	00792	00793	00794	00795	00796	00797	00798	00799
01440	00800	00801	00802	00803	00804	00805	00806	00807
01450	00808	00809	00810	00811	00812	00813	00814	00815
01460	00816	00817	00818	00819	00820	00821	00822	00823
01470	00824	00825	00826	00827	00828	00829	00830	00831
01500	00832	00833	00834	00835	00836	00837	00838	00839
01510	00840	00841	00842	00843	00844	00845	00846	00847
01520	00848	00849	00850	00851	00852	00853	00854	00855
01530	00856	00857	00858	00859	00860	00861	00862	00863
01540	00864	00865	00866	00867	00868	00869	00870	00871
01550	00872	00873	00874	00875	00876	00877	00878	00879
01560	00880	00881	00882	00883	00884	00885	00886	00887
01570	00888	00889	00890	00891	00892	00893	00894	00895
01600	00896	00897	00898	00899	00900	00901	00902	00903
01610	00904	00905	00906	00907	00908	00909	00910	00911
01620	00912	00913	00914	00915	00916	00917	00918	00919
01630	00920	00921	00922	00923	00924	00925	00926	00927
01640	00928	00929	00930	00931	00932	00933	00934	00935
01650	00936	00937	00938	00939	00940	00941	00942	00943
01660	00944	00945	00946	00947	00948	00949	00950	00951
01670	00952	00953	00954	00955	00956	00957	00958	00959
01700	00960	00961	00962	00963	00964	00965	00966	00967
01710	00968	00969	00970	00971	00972	00973	00974	00975
01720	00976	00977	00978	00979	00980	00981	00982	00983
01730	00984	00985	00986	00987	00988	00989	00990	00991
01740	00992	00993	00994	00995	00996	00997	00998	00999
01750	01000	01001	01002	01003	01004	01005	01006	01007
01760	01008	01009	01010	01011	01012	01013	01014	01015
01770	01016	01017	01018	01019	01020	01021	01022	01023

OCTAL	0	1	2	3	4	5	6	7
02000	01024	01025	01026	01027	01028	01029	01030	01031
02010	01032	01033	01034	01035	01036	01037	01038	01039
02020	01040	01041	01042	01043	01044	01045	01046	01047
02030	01048	01049	01050	01051	01052	01053	01054	01055
02040	01056	01057	01058	01059	01060	01061	01062	01063
02050	01064	01065	01066	01067	01068	01069	01070	01071
02060	01072	01073	01074	01075	01076	01077	01078	01079
02070	01080	01081	01082	01083	01084	01085	01086	01087
02100	01088	01089	01090	01091	01092	01093	01094	01095
02110	01096	01097	01098	01099	01100	01101	01102	01103
02120	01104	01105	01106	01107	01108	01109	01110	01111
02130	01112	01113	01114	01115	01116	01117	01118	01119
02140	01120	01121	01122	01123	01124	01125	01126	01127
02150	01128	01129	01130	01131	01132	01133	01134	01135
02160	01136	01137	01138	01139	01140	01141	01142	01143
02170	01144	01145	01146	01147	01148	01149	01150	01151
02200	01152	01153	01154	01155	01156	01157	01158	01159
02210	01160	01161	01162	01163	01164	01165	01166	01167
02220	01168	01169	01170	01171	01172	01173	01174	01175
02230	01176	01177	01178	01179	01180	01181	01182	01183
02240	01184	01185	01186	01187	01188	01189	01190	01191
02250	01192	01193	01194	01195	01196	01197	01198	01199
02260	01200	01201	01202	01203	01204	01205	01206	01207
02270	01208	01209	01210	01211	01212	01213	01214	01215
02300	01216	01217	01218	01219	01220	01221	01222	01223
02310	01224	01225	01226	01227	01228	01229	01230	01231
02320	01232	01233	01234	01235	01236	01237	01238	01239
02330	01240	01241	01242	01243	01244	01245	01246	01247
02340	01248	01249	01250	01251	01252	01253	01254	01255
02350	01256	01257	01258	01259	01260	01261	01262	01263
02360	01264	01265	01266	01267	01268	01269	01270	01271
02370	01272	01273	01274	01275	01276	01277	01278	01279
02400	01280	01281	01282	01283	01284	01285	01286	01287
02410	01288	01289	01290	01291	01292	01293	01294	01295
02420	01296	01297	01298	01299	01300	01301	01302	01303
02430	01304	01305	01306	01307	01308	01309	01310	01311
02440	01312	01313	01314	01315	01316	01317	01318	01319
02450	01320	01321	01322	01323	01324	01325	01326	01327
02460	01328	01329	01330	01331	01332	01333	01334	01335
02470	01336	01337	01338	01339	01340	01341	01342	01343
02500	01344	01345	01346	01347	01348	01349	01350	01351
02510	01352	01353	01354	01355	01356	01357	01358	01359
02520	01360	01361	01362	01363	01364	01365	01366	01367
02530	01368	01369	01370	01371	01372	01373	01374	01375
02540	01376	01377	01378	01379	01380	01381	01382	01383
02550	01384	01385	01386	01387	01388	01389	01390	01391
02560	01392	01393	01394	01395	01396	01397	01398	01399
02570	01400	01401	01402	01403	01404	01405	01406	01407
02600	01408	01409	01410	01411	01412	01413	01414	01415
02610	01416	01417	01418	01419	01420	01421	01422	01423
02620	01424	01425	01426	01427	01428	01429	01430	01431
02630	01432	01433	01434	01435	01436	01437	01438	01439
02640	01440	01441	01442	01443	01444	01445	01446	01447
02650	01448	01449	01450	01451	01452	01453	01454	01455
02660	01456	01457	01458	01459	01460	01461	01462	01463
02670	01464	01465	01466	01467	01468	01469	01470	01471
02700	01472	01473	01474	01475	01476	01477	01478	01479
02710	01480	01481	01482	01483	01484	01485	01486	01487
02720	01488	01489	01490	01491	01492	01493	01494	01495
02730	01496	01497	01498	01499	01500	01501	01502	01503
02740	01504	01505	01506	01507	01508	01509	01510	01511
02750	01512	01513	01514	01515	01516	01517	01518	01519
02760	01520	01521	01522	01523	01524	01525	01526	01527
02770	01528	01529	01530	01531	01532	01533	01534	01535

OCTAL	0	1	2	3	4	5	6	7
03000	01536	01537	01538	01539	01540	01541	01542	01543
03010	01544	01545	01546	01547	01548	01549	01550	01551
03020	01552	01553	01554	01555	01556	01557	01558	01559
03030	01560	01561	01562	01563	01564	01565	01566	01567
03040	01568	01569	01570	01571	01572	01573	01574	01575
03050	01576	01577	01578	01579	01580	01581	01582	01583
03060	01584	01585	01586	01587	01588	01589	01590	01591
03070	01592	01593	01594	01595	01596	01597	01598	01599
03100	01600	01601	01602	01603	01604	01605	01606	01607
03110	01608	01609	01610	01611	01612	01613	01614	01615
03120	01616	01617	01618	01619	01620	01621	01622	01623
03130	01624	01625	01626	01627	01628	01629	01630	01631
03140	01632	01633	01634	01635	01636	01637	01638	01639
03150	01640	01641	01642	01643	01644	01645	01646	01647
03160	01648	01649	01650	01651	01652	01653	01654	01655
03170	01656	01657	01658	01659	01660	01661	01662	01663
03200	01664	01665	01666	01667	01668	01669	01670	01671
03210	01672	01673	01674	01675	01676	01677	01678	01679
03220	01680	01681	01682	01683	01684	01685	01686	01687
03230	01688	01689	01690	01691	01692	01693	01694	01695
03240	01696	01697	01698	01699	01700	01701	01702	01703
03250	01704	01705	01706	01707	01708	01709	01710	01711
03260	01712	01713	01714	01715	01716	01717	01718	01719
03270	01720	01721	01722	01723	01724	01725	01726	01727
03300	01728	01729	01730	01731	01732	01733	01734	01735
03310	01736	01737	01738	01739	01740	01741	01742	01743
03320	01744	01745	01746	01747	01748	01749	01750	01751
03330	01752	01753	01754	01755	01756	01757	01758	01759
03340	01760	01761	01762	01763	01764	01765	01766	01767
03350	01768	01769	01770	01771	01772	01773	01774	01775
03360	01776	01777	01778	01779	01780	01781	01782	01783
03370	01784	01785	01786	01787	01788	01789	01790	01791
03400	01792	01793	01794	01795	01796	01797	01798	01799
03410	01800	01801	01802	01803	01804	01805	01806	01807
03420	01808	01809	01810	01811	01812	01813	01814	01815
03430	01816	01817	01818	01819	01820	01821	01822	01823
03440	01824	01825	01826	01827	01828	01829	01830	01831
03450	01832	01833	01834	01835	01836	01837	01838	01839
03460	01840	01841	01842	01843	01844	01845	01846	01847
03470	01848	01849	01850	01851	01852	01853	01854	01855
03500	01856	01857	01858	01859	01860	01861	01862	01863
03510	01864	01865	01866	01867	01868	01869	01870	01871
03520	01872	01873	01874	01875	01876	01877	01878	01879
03530	01880	01881	01882	01883	01884	01885	01886	01887
03540	01888	01889	01890	01891	01892	01893	01894	01895
03550	01896	01897	01898	01899	01900	01901	01902	01903
03560	01904	01905	01906	01907	01908	01909	01910	01911
03570	01912	01913	01914	01915	01916	01917	01918	01919
03600	01920	01921	01922	01923	01924	01925	01926	01927
03610	01928	01929	01930	01931	01932	01933	01934	01935
03620	01936	01937	01938	01939	01940	01941	01942	01943
03630	01944	01945	01946	01947	01948	01949	01950	01951
03640	01952	01953	01954	01955	01956	01957	01958	01959
03650	01960	01961	01962	01963	01964	01965	01966	01967
03660	01968	01969	01970	01971	01972	01973	01974	01975
03670	01976	01977	01978	01979	01980	01981	01982	01983
03700	01984	01985	01986	01987	01988	01989	01990	01991
03710	01992	01993	01994	01995	01996	01997	01998	01999
03720	02000	02001	02002	02003	02004	02005	02006	02007
03730	02008	02009	02010	02011	02012	02013	02014	02015
03740	02016	02017	02018	02019	02020	02021	02022	02023
03750	02024	02025	02026	02027	02028	02029	02030	02031
03760	02032	02033	02034	02035	02036	02037	02038	02039
03770	02040	02041	02042	02043	02044	02045	02046	02047

OCTAL	0	1	2	3	4	5	6	7
04000	02048	02049	02050	02051	02052	02053	02054	02055
04010	02056	02057	02058	02059	02060	02061	02062	02063
04020	02064	02065	02066	02067	02068	02069	02070	02071
04030	02072	02073	02074	02075	02076	02077	02078	02079
04040	02080	02081	02082	02083	02084	02085	02086	02087
04050	02088	02089	02090	02091	02092	02093	02094	02095
04060	02096	02097	02098	02099	02100	02101	02102	02103
04070	02104	02105	02106	02107	02108	02109	02110	02111
04100	02112	02113	02114	02115	02116	02117	02118	02119
04110	02120	02121	02122	02123	02124	02125	02126	02127
04120	02128	02129	02130	02131	02132	02133	02134	02135
04130	02136	02137	02138	02139	02140	02141	02142	02143
04140	02144	02145	02146	02147	02148	02149	02150	02151
04150	02152	02153	02154	02155	02156	02157	02158	02159
04160	02160	02161	02162	02163	02164	02165	02166	02167
04170	02168	02169	02170	02171	02172	02173	02174	02175
04200	02176	02177	02178	02179	02180	02181	02182	02183
04210	02184	02185	02186	02187	02188	02189	02190	02191
04220	02192	02193	02194	02195	02196	02197	02198	02199
04230	02200	02201	02202	02203	02204	02205	02206	02207
04240	02208	02209	02210	02211	02212	02213	02214	02215
04250	02216	02217	02218	02219	02220	02221	02222	02223
04260	02224	02225	02226	02227	02228	02229	02230	02231
04270	02232	02233	02234	02235	02236	02237	02238	02239
04300	02240	02241	02242	02243	02244	02245	02246	02247
04310	02248	02249	02250	02251	02252	02253	02254	02255
04320	02256	02257	02258	02259	02260	02261	02262	02263
04330	02264	02265	02266	02267	02268	02269	02270	02271
04340	02272	02273	02274	02275	02276	02277	02278	02279
04350	02280	02281	02282	02283	02284	02285	02286	02287
04360	02288	02289	02290	02291	02292	02293	02294	02295
04370	02296	02297	02298	02299	02300	02301	02302	02303
04400	02304	02305	02306	02307	02308	02309	02310	02311
04410	02312	02313	02314	02315	02316	02317	02318	02319
04420	02320	02321	02322	02323	02324	02325	02326	02327
04430	02328	02329	02330	02331	02332	02333	02334	02335
04440	02336	02337	02338	02339	02340	02341	02342	02343
04450	02344	02345	02346	02347	02348	02349	02350	02351
04460	02352	02353	02354	02355	02356	02357	02358	02359
04470	02360	02361	02362	02363	02364	02365	02366	02367
04500	02368	02369	02370	02371	02372	02373	02374	02375
04510	02376	02377	02378	02379	02380	02381	02382	02383
04520	02384	02385	02386	02387	02388	02389	02390	02391
04530	02392	02393	02394	02395	02396	02397	02398	02399
04540	02400	02401	02402	02403	02404	02405	02406	02407
04550	02408	02409	02410	02411	02412	02413	02414	02415
04560	02416	02417	02418	02419	02420	02421	02422	02423
04570	02424	02425	02426	02427	02428	02429	02430	02431
04600	02432	02433	02434	02435	02436	02437	02438	02439
04610	02440	02441	02442	02443	02444	02445	02446	02447
04620	02448	02449	02450	02451	02452	02453	02454	02455
04630	02456	02457	02458	02459	02460	02461	02462	02463
04640	02464	02465	02466	02467	02468	02469	02470	02471
04650	02472	02473	02474	02475	02476	02477	02478	02479
04660	02480	02481	02482	02483	02484	02485	02486	02487
04670	02488	02489	02490	02491	02492	02493	02494	02495
04700	02496	02497	02498	02499	02500	02501	02502	02503
04710	02504	02505	02506	02507	02508	02509	02510	02511
04720	02512	02513	02514	02515	02516	02517	02518	02519
04730	02520	02521	02522	02523	02524	02525	02526	02527
04740	02528	02529	02530	02531	02532	02533	02534	02535
04750	02536	02537	02538	02539	02540	02541	02542	02543
04760	02544	02545	02546	02547	02548	02549	02550	02551
04770	02552	02553	02554	02555	02556	02557	02558	02559

OCTAL	0	1	2	3	4	5	6	7
05000	02560	02561	02562	02563	02564	02565	02566	02567
05010	02568	02569	02570	02571	02572	02573	02574	02575
05020	02576	02577	02578	02579	02580	02581	02582	02583
05030	02584	02585	02586	02587	02588	02589	02590	02591
05040	02592	02593	02594	02595	02596	02597	02598	02599
05050	02600	02601	02602	02603	02604	02605	02606	02607
05060	02608	02609	02610	02611	02612	02613	02614	02615
05070	02616	02617	02618	02619	02620	02621	02622	02623
05100	02624	02625	02626	02627	02628	02629	02630	02631
05110	02632	02633	02634	02635	02636	02637	02638	02639
05120	02640	02641	02642	02643	02644	02645	02646	02647
05130	02648	02649	02650	02651	02652	02653	02654	02655
05140	02656	02657	02658	02659	02660	02661	02662	02663
05150	02664	02665	02666	02667	02668	02669	02670	02671
05160	02672	02673	02674	02675	02676	02677	02678	02679
05170	02680	02681	02682	02683	02684	02685	02686	02687
05200	02688	02689	02690	02691	02692	02693	02694	02695
05210	02696	02697	02698	02699	02700	02701	02702	02703
05220	02704	02705	02706	02707	02708	02709	02710	02711
05230	02712	02713	02714	02715	02716	02717	02718	02719
05240	02720	02721	02722	02723	02724	02725	02726	02727
05250	02728	02729	02730	02731	02732	02733	02734	02735
05260	02736	02737	02738	02739	02740	02741	02742	02743
05270	02744	02745	02746	02747	02748	02749	02750	02751
05300	02752	02753	02754	02755	02756	02757	02758	02759
05310	02760	02761	02762	02763	02764	02765	02766	02767
05320	02768	02769	02770	02771	02772	02773	02774	02775
05330	02776	02777	02778	02779	02780	02781	02782	02783
05340	02784	02785	02786	02787	02788	02789	02790	02791
05350	02792	02793	02794	02795	02796	02797	02798	02799
05360	02800	02801	02802	02803	02804	02805	02806	02807
05370	02808	02809	02810	02811	02812	02813	02814	02815
05400	02816	02817	02818	02819	02820	02821	02822	02823
05410	02824	02825	02826	02827	02828	02829	02830	02831
05420	02832	02833	02834	02835	02836	02837	02838	02839
05430	02840	02841	02842	02843	02844	02845	02846	02847
05440	02848	02849	02850	02851	02852	02853	02854	02855
05450	02856	02857	02858	02859	02860	02861	02862	02863
05460	02864	02865	02866	02867	02868	02869	02870	02871
05470	02872	02873	02874	02875	02876	02877	02878	02879
05500	02880	02881	02882	02883	02884	02885	02886	02887
05510	02888	02889	02890	02891	02892	02893	02894	02895
05520	02896	02897	02898	02899	02900	02901	02902	02903
05530	02904	02905	02906	02907	02908	02909	02910	02911
05540	02912	02913	02914	02915	02916	02917	02918	02919
05550	02920	02921	02922	02923	02924	02925	02926	02927
05560	02928	02929	02930	02931	02932	02933	02934	02935
05570	02936	02937	02938	02939	02940	02941	02942	02943
05600	02944	02945	02946	02947	02948	02949	02950	02951
05610	02952	02953	02954	02955	02956	02957	02958	02959
05620	02960	02961	02962	02963	02964	02965	02966	02967
05630	02968	02969	02970	02971	02972	02973	02974	02975
05640	02976	02977	02978	02979	02980	02981	02982	02983
05650	02984	02985	02986	02987	02988	02989	02990	02991
05660	02992	02993	02994	02995	02996	02997	02998	02999
05670	03000	03001	03002	03003	03004	03005	03006	03007
05700	03008	03009	03010	03011	03012	03013	03014	03015
05710	03016	03017	03018	03019	03020	03021	03022	03023
05720	03024	03025	03026	03027	03028	03029	03030	03031
05730	03032	03033	03034	03035	03036	03037	03038	03039
05740	03040	03041	03042	03043	03044	03045	03046	03047
05750	03048	03049	03050	03051	03052	03053	03054	03055
05760	03056	03057	03058	03059	03060	03061	03062	03063
05770	03064	03065	03066	03067	03068	03069	03070	03071

OCTAL	0	1	2	3	4	5	6	7
06000	03072	03073	03074	03075	03076	03077	03078	03079
06010	03080	03081	03082	03083	03084	03085	03086	03087
06020	03088	03089	03090	03091	03092	03093	03094	03095
06030	03096	03097	03098	03099	03100	03101	03102	03103
06040	03104	03105	03106	03107	03108	03109	03110	03111
06050	03112	03113	03114	03115	03116	03117	03118	03119
06060	03120	03121	03122	03123	03124	03125	03126	03127
06070	03128	03129	03130	03131	03132	03133	03134	03135
06100	03136	03137	03138	03139	03140	03141	03142	03143
06110	03144	03145	03146	03147	03148	03149	03150	03151
06120	03152	03153	03154	03155	03156	03157	03158	03159
06130	03160	03161	03162	03163	03164	03165	03166	03167
06140	03168	03169	03170	03171	03172	03173	03174	03175
06150	03176	03177	03178	03179	03180	03181	03182	03183
06160	03184	03185	03186	03187	03188	03189	03190	03191
06170	03192	03193	03194	03195	03196	03197	03198	03199
06200	03200	03201	03202	03203	03204	03205	03206	03207
06210	03208	03209	03210	03211	03212	03213	03214	03215
06220	03216	03217	03218	03219	03220	03221	03222	03223
06230	03224	03225	03226	03227	03228	03229	03230	03231
06240	03232	03233	03234	03235	03236	03237	03238	03239
06250	03240	03241	03242	03243	03244	03245	03246	03247
06260	03248	03249	03250	03251	03252	03253	03254	03255
06270	03256	03257	03258	03259	03260	03261	03262	03263
06300	03264	03265	03266	03267	03268	03269	03270	03271
06310	03272	03273	03274	03275	03276	03277	03278	03279
06320	03280	03281	03282	03283	03284	03285	03286	03287
06330	03288	03289	03290	03291	03292	03293	03294	03295
06340	03296	03297	03298	03299	03300	03301	03302	03303
06350	03304	03305	03306	03307	03308	03309	03310	03311
06360	03312	03313	03314	03315	03316	03317	03318	03319
06370	03320	03321	03322	03323	03324	03325	03326	03327
06400	03328	03329	03330	03331	03332	03333	03334	03335
06410	03336	03337	03338	03339	03340	03341	03342	03343
06420	03344	03345	03346	03347	03348	03349	03350	03351
06430	03352	03353	03354	03355	03356	03357	03358	03359
06440	03360	03361	03362	03363	03364	03365	03366	03367
06450	03368	03369	03370	03371	03372	03373	03374	03375
06460	03376	03377	03378	03379	03380	03381	03382	03383
06470	03384	03385	03386	03387	03388	03389	03390	03391
06500	03392	03393	03394	03395	03396	03397	03398	03399
06510	03400	03401	03402	03403	03404	03405	03406	03407
06520	03408	03409	03410	03411	03412	03413	03414	03415
06530	03416	03417	03418	03419	03420	03421	03422	03423
06540	03424	03425	03426	03427	03428	03429	03430	03431
06550	03432	03433	03434	03435	03436	03437	03438	03439
06560	03440	03441	03442	03443	03444	03445	03446	03447
06570	03448	03449	03450	03451	03452	03453	03454	03455
06600	03456	03457	03458	03459	03460	03461	03462	03463
06610	03464	03465	03466	03467	03468	03469	03470	03471
06620	03472	03473	03474	03475	03476	03477	03478	03479
06630	03480	03481	03482	03483	03484	03485	03486	03487
06640	03488	03489	03490	03491	03492	03493	03494	03495
06650	03496	03497	03498	03499	03500	03501	03502	03503
06660	03504	03505	03506	03507	03508	03509	03510	03511
06670	03512	03513	03514	03515	03516	03517	03518	03519
06700	03520	03521	03522	03523	03524	03525	03526	03527
06710	03528	03529	03530	03531	03532	03533	03534	03535
06720	03536	03537	03538	03539	03540	03541	03542	03543
06730	03544	03545	03546	03547	03548	03549	03550	03551
06740	03552	03553	03554	03555	03556	03557	03558	03559
06750	03560	03561	03562	03563	03564	03565	03566	03567
06760	03568	03569	03570	03571	03572	03573	03574	03575
06770	03576	03577	03578	03579	03580	03581	03582	03583

OCTAL	0	1	2	3	4	5	6	7
07000	03584	03585	03586	03587	03588	03589	03590	03591
07010	03592	03593	03594	03595	03596	03597	03598	03599
07020	03600	03601	03602	03603	03604	03605	03606	03607
07030	03608	03609	03610	03611	03612	03613	03614	03615
07040	03616	03617	03618	03619	03620	03621	03622	03623
07050	03624	03625	03626	03627	03628	03629	03630	03631
07060	03632	03633	03634	03635	03636	03637	03638	03639
07070	03640	03641	03642	03643	03644	03645	03646	03647
07100	03648	03649	03650	03651	03652	03653	03654	03655
07110	03656	03657	03658	03659	03660	03661	03662	03663
07120	03664	03665	03666	03667	03668	03669	03670	03671
07130	03672	03673	03674	03675	03676	03677	03678	03679
07140	03680	03681	03682	03683	03684	03685	03686	03687
07150	03688	03689	03690	03691	03692	03693	03694	03695
07160	03696	03697	03698	03699	03700	03701	03702	03703
07170	03704	03705	03706	03707	03708	03709	03710	03711
07200	03712	03713	03714	03715	03716	03717	03718	03719
07210	03720	03721	03722	03723	03724	03725	03726	03727
07220	03728	03729	03730	03731	03732	03733	03734	03735
07230	03736	03737	03738	03739	03740	03741	03742	03743
07240	03744	03745	03746	03747	03748	03749	03750	03751
07250	03752	03753	03754	03755	03756	03757	03758	03759
07260	03760	03761	03762	03763	03764	03765	03766	03767
07270	03768	03769	03770	03771	03772	03773	03774	03775
07300	03776	03777	03778	03779	03780	03781	03782	03783
07310	03784	03785	03786	03787	03788	03789	03790	03791
07320	03792	03793	03794	03795	03796	03797	03798	03799
07330	03800	03801	03802	03803	03804	03805	03806	03807
07340	03808	03809	03810	03811	03812	03813	03814	03815
07350	03816	03817	03818	03819	03820	03821	03822	03823
07360	03824	03825	03826	03827	03828	03829	03830	03831
07370	03832	03833	03834	03835	03836	03837	03838	03839
07400	03840	03841	03842	03843	03844	03845	03846	03847
07410	03848	03849	03850	03851	03852	03853	03854	03855
07420	03856	03857	03858	03859	03860	03861	03862	03863
07430	03864	03865	03866	03867	03868	03869	03870	03871
07440	03872	03873	03874	03875	03876	03877	03878	03879
07450	03880	03881	03882	03883	03884	03885	03886	03887
07460	03888	03889	03890	03891	03892	03893	03894	03895
07470	03896	03897	03898	03899	03900	03901	03902	03903
07500	03904	03905	03906	03907	03908	03909	03910	03911
07510	03912	03913	03914	03915	03916	03917	03918	03919
07520	03920	03921	03922	03923	03924	03925	03926	03927
07530	03928	03929	03930	03931	03932	03933	03934	03935
07540	03936	03937	03938	03939	03940	03941	03942	03943
07550	03944	03945	03946	03947	03948	03949	03950	03951
07560	03952	03953	03954	03955	03956	03957	03958	03959
07570	03960	03961	03962	03963	03964	03965	03966	03967
07600	03968	03969	03970	03971	03972	03973	03974	03975
07610	03976	03977	03978	03979	03980	03981	03982	03983
07620	03984	03985	03986	03987	03988	03989	03990	03991
07630	03992	03993	03994	03995	03996	03997	03998	03999
07640	04000	04001	04002	04003	04004	04005	04006	04007
07650	04008	04009	04010	04011	04012	04013	04014	04015
07660	04016	04017	04018	04019	04020	04021	04022	04023
07670	04024	04025	04026	04027	04028	04029	04030	04031
07700	04032	04033	04034	04035	04036	04037	04038	04039
07710	04040	04041	04042	04043	04044	04045	04046	04047
07720	04048	04049	04050	04051	04052	04053	04054	04055
07730	04056	04057	04058	04059	04060	04061	04062	04063
07740	04064	04065	04066	04067	04068	04069	04070	04071
07750	04072	04073	04074	04075	04076	04077	04078	04079
07760	04080	04081	04082	04083	04084	04085	04086	04087
07770	04088	04089	04090	04091	04092	04093	04094	04095

OCTAL	0	1	2	3	4	5	6	7
10000	04096	04097	04098	04099	04100	04101	04102	04103
10010	04104	04105	04106	04107	04108	04109	04110	04111
10020	04112	04113	04114	04115	04116	04117	04118	04119
10030	04120	04121	04122	04123	04124	04125	04126	04127
10040	04128	04129	04130	04131	04132	04133	04134	04135
10050	04136	04137	04138	04139	04140	04141	04142	04143
10060	04144	04145	04146	04147	04148	04149	04150	04151
10070	04152	04153	04154	04155	04156	04157	04158	04159
10100	04160	04161	04162	04163	04164	04165	04166	04167
10110	04168	04169	04170	04171	04172	04173	04174	04175
10120	04176	04177	04178	04179	04180	04181	04182	04183
10130	04184	04185	04186	04187	04188	04189	04190	04191
10140	04192	04193	04194	04195	04196	04197	04198	04199
10150	04200	04201	04202	04203	04204	04205	04206	04207
10160	04208	04209	04210	04211	04212	04213	04214	04215
10170	04216	04217	04218	04219	04220	04221	04222	04223
10200	04224	04225	04226	04227	04228	04229	04230	04231
10210	04232	04233	04234	04235	04236	04237	04238	04239
10220	04240	04241	04242	04243	04244	04245	04246	04247
10230	04248	04249	04250	04251	04252	04253	04254	04255
10240	04256	04257	04258	04259	04260	04261	04262	04263
10250	04264	04265	04266	04267	04268	04269	04270	04271
10260	04272	04273	04274	04275	04276	04277	04278	04279
10270	04280	04281	04282	04283	04284	04285	04286	04287
10300	04288	04289	04290	04291	04292	04293	04294	04295
10310	04296	04297	04298	04299	04300	04301	04302	04303
10320	04304	04305	04306	04307	04308	04309	04310	04311
10330	04312	04313	04314	04315	04316	04317	04318	04319
10340	04320	04321	04322	04323	04324	04325	04326	04327
10350	04328	04329	04330	04331	04332	04333	04334	04335
10360	04336	04337	04338	04339	04340	04341	04342	04343
10370	04344	04345	04346	04347	04348	04349	04350	04351
10400	04352	04353	04354	04355	04356	04357	04358	04359
10410	04360	04361	04362	04363	04364	04365	04366	04367
10420	04368	04369	04370	04371	04372	04373	04374	04375
10430	04376	04377	04378	04379	04380	04381	04382	04383
10440	04384	04385	04386	04387	04388	04389	04390	04391
10450	04392	04393	04394	04395	04396	04397	04398	04399
10460	04400	04401	04402	04403	04404	04405	04406	04407
10470	04408	04409	04410	04411	04412	04413	04414	04415
10500	04416	04417	04418	04419	04420	04421	04422	04423
10510	04424	04425	04426	04427	04428	04429	04430	04431
10520	04432	04433	04434	04435	04436	04437	04438	04439
10530	04440	04441	04442	04443	04444	04445	04446	04447
10540	04446	04449	04450	04451	04452	04453	04454	04455
10550	04456	04457	04458	04459	04460	04461	04462	04463
10560	04464	04465	04466	04467	04468	04469	04470	04471
10570	04472	04473	04474	04475	04476	04477	04478	04479
10600	04480	04481	04482	04483	04484	04485	04486	04487
10610	04488	04489	04490	04491	04492	04493	04494	04495
10620	04496	04497	04498	04499	04500	04501	04502	04503
10630	04504	04505	04506	04507	04508	04509	04510	04511
10640	04512	04513	04514	04515	04516	04517	04518	04519
10650	04520	04521	04522	04523	04524	04525	04526	04527
10660	04528	04529	04530	04531	04532	04533	04534	04535
10670	04536	04537	04538	04539	04540	04541	04542	04543
10700	04544	04545	04546	04547	04548	04549	04550	04551
10710	04552	04553	04554	04555	04556	04557	04558	04559
10720	04560	04561	04562	04563	04564	04565	04566	04567
10730	04568	04569	04570	04571	04572	04573	04574	04575
10740	04576	04577	04578	04579	04580	04581	04582	04583
10750	04584	04585	04586	04587	04588	04589	04590	04591
10760	04592	04593	04594	04595	04596	04597	04598	04599
10770	04600	04601	04602	04603	04604	04605	04606	04607

OCTAL	0	1	2	3	4	5	6	7
11000	04608	04609	04610	04611	04612	04613	04614	04615
11010	04616	04617	04618	04619	04620	04621	04622	04623
11020	04624	04625	04626	04627	04628	04629	04630	04631
11030	04632	04633	04634	04635	04636	04637	04638	04639
11040	04640	04641	04642	04643	04644	04645	04646	04647
11050	04648	04649	04650	04651	04652	04653	04654	04655
11060	04656	04657	04658	04659	04660	04661	04662	04663
11070	04664	04665	04666	04667	04668	04669	04670	04671
11100	04672	04673	04674	04675	04676	04677	04678	04679
11110	04680	04681	04682	04683	04684	04685	04686	04687
11120	04688	04689	04690	04691	04692	04693	04694	04695
11130	04696	04697	04698	04699	04700	04701	04702	04703
11140	04704	04705	04706	04707	04708	04709	04710	04711
11150	04712	04713	04714	04715	04716	04717	04718	04719
11160	04720	04721	04722	04723	04724	04725	04726	04727
11170	04728	04729	04730	04731	04732	04733	04734	04735
11200	04736	04737	04738	04739	04740	04741	04742	04743
11210	04744	04745	04746	04747	04748	04749	04750	04751
11220	04752	04753	04754	04755	04756	04757	04758	04759
11230	04760	04761	04762	04763	04764	04765	04766	04767
11240	04768	04769	04770	04771	04772	04773	04774	04775
11250	04776	04777	04778	04779	04780	04781	04782	04783
11260	04784	04785	04786	04787	04788	04789	04790	04791
11270	04792	04793	04794	04795	04796	04797	04798	04799
11300	04800	04801	04802	04803	04804	04805	04806	04807
11310	04808	04809	04810	04811	04812	04813	04814	04815
11320	04816	04817	04818	04819	04820	04821	04822	04823
11330	04824	04825	04826	04827	04828	04829	04830	04831
11340	04832	04833	04834	04835	04836	04837	04838	04839
11350	04840	04841	04842	04843	04844	04845	04846	04847
11360	04848	04849	04850	04851	04852	04853	04854	04855
11370	04856	04857	04858	04859	04860	04861	04862	04863
11400	04864	04865	04866	04867	04868	04869	04870	04871
11410	04872	04873	04874	04875	04876	04877	04878	04879
11420	04880	04881	04882	04883	04884	04885	04886	04887
11430	04888	04889	04890	04891	04892	04893	04894	04895
11440	04896	04897	04898	04899	04900	04901	04902	04903
11450	04904	04905	04906	04907	04908	04909	04910	04911
11460	04912	04913	04914	04915	04916	04917	04918	04919
11470	04920	04921	04922	04923	04924	04925	04926	04927
11500	04928	04929	04930	04931	04932	04933	04934	04935
11510	04936	04937	04938	04939	04940	04941	04942	04943
11520	04944	04945	04946	04947	04948	04949	04950	04951
11530	04952	04953	04954	04955	04956	04957	04958	04959
11540	04960	04961	04962	04963	04964	04965	04966	04967
11550	04968	04969	04970	04971	04972	04973	04974	04975
11560	04976	04977	04978	04979	04980	04981	04982	04983
11570	04984	04985	04986	04987	04988	04989	04990	04991
11600	04992	04993	04994	04995	04996	04997	04998	04999
11610	05000	05001	05002	05003	05004	05005	05006	05007
11620	05008	05009	05010	05011	05012	05013	05014	05015
11630	05016	05017	05018	05019	05020	05021	05022	05023
11640	05024	05025	05026	05027	05028	05029	05030	05031
11650	05032	05033	05034	05035	05036	05037	05038	05039
11660	05040	05041	05042	05043	05044	05045	05046	05047
11670	05048	05049	05050	05051	05052	05053	05054	05055
11700	05056	05057	05058	05059	05060	05061	05062	05063
11710	05064	05065	05066	05067	05068	05069	05070	05071
11720	05072	05073	05074	05075	05076	05077	05078	05079
11730	05080	05081	05082	05083	05084	05085	05086	05087
11740	05088	05089	05090	05091	05092	05093	05094	05095
11750	05096	05097	05098	05099	05100	05101	05102	05103
11760	05104	05105	05106	05107	05108	05109	05110	05111
11770	05112	05113	05114	05115	05116	05117	05118	05119

OCTAL	0	1	2	3	4	5	6	7
12000	05120	05121	05122	05123	05124	05125	05126	05127
12010	05128	05129	05130	05131	05132	05133	05134	05135
12020	05136	05137	05138	05139	05140	05141	05142	05143
12030	05144	05145	05146	05147	05148	05149	05150	05151
12040	05152	05153	05154	05155	05156	05157	05158	05159
12050	05160	05161	05162	05163	05164	05165	05166	05167
12060	05168	05169	05170	05171	05172	05173	05174	05175
12070	05176	05177	05178	05179	05180	05181	05182	05183
12100	05184	05185	05186	05187	05188	05189	05190	05191
12110	05192	05193	05194	05195	05196	05197	05198	05199
12120	05200	05201	05202	05203	05204	05205	05206	05207
12130	05208	05209	05210	05211	05212	05213	05214	05215
12140	05216	05217	05218	05219	05220	05221	05222	05223
12150	05224	05225	05226	05227	05228	05229	05230	05231
12160	05232	05233	05234	05235	05236	05237	05238	05239
12170	05240	05241	05242	05243	05244	05245	05246	05247
12200	05248	05249	05250	05251	05252	05253	05254	05255
12210	05256	05257	05258	05259	05260	05261	05262	05263
12220	05264	05265	05266	05267	05268	05269	05270	05271
12230	05272	05273	05274	05275	05276	05277	05278	05279
12240	05280	05281	05282	05283	05284	05285	05286	05287
12250	05288	05289	05290	05291	05292	05293	05294	05295
12260	05296	05297	05298	05299	05300	05301	05302	05303
12270	05304	05305	05306	05307	05308	05309	05310	05311
12300	05312	05313	05314	05315	05316	05317	05318	05319
12310	05320	05321	05322	05323	05324	05325	05326	05327
12320	05328	05329	05330	05331	05332	05333	05334	05335
12330	05336	05337	05338	05339	05340	05341	05342	05343
12340	05344	05345	05346	05347	05348	05349	05350	05351
12350	05352	05353	05354	05355	05356	05357	05358	05359
12360	05360	05361	05362	05363	05364	05365	05366	05367
12370	05368	05369	05370	05371	05372	05373	05374	05375
12400	05376	05377	05378	05379	05380	05381	05382	05383
12410	05384	05385	05386	05387	05388	05389	05390	05391
12420	05392	05393	05394	05395	05396	05397	05398	05399
12430	05400	05401	05402	05403	05404	05405	05406	05407
12440	05408	05409	05410	05411	05412	05413	05414	05415
12450	05416	05417	05418	05419	05420	05421	05422	05423
12460	05424	05425	05426	05427	05428	05429	05430	05431
12470	05432	05433	05434	05435	05436	05437	05438	05439
12500	05440	05441	05442	05443	05444	05445	05446	05447
12510	05448	05449	05450	05451	05452	05453	05454	05455
12520	05456	05457	05458	05459	05460	05461	05462	05463
12530	05464	05465	05466	05467	05468	05469	05470	05471
12540	05472	05473	05474	05475	05476	05477	05478	05479
12550	05480	05481	05482	05483	05484	05485	05486	05487
12560	05488	05489	05490	05491	05492	05493	05494	05495
12570	05496	05497	05498	05499	05500	05501	05502	05503
12600	05504	05505	05506	05507	05508	05509	05510	05511
12610	05512	05513	05514	05515	05516	05517	05518	05519
12620	05520	05521	05522	05523	05524	05525	05526	05527
12630	05528	05529	05530	05531	05532	05533	05534	05535
12640	05536	05537	05538	05539	05540	05541	05542	05543
12650	05544	05545	05546	05547	05548	05549	05550	05551
12660	05552	05553	05554	05555	05556	05557	05558	05559
12670	05560	05561	05562	05563	05564	05565	05566	05567
12700	05568	05569	05570	05571	05572	05573	05574	05575
12710	05576	05577	05578	05579	05580	05581	05582	05583
12720	05584	05585	05586	05587	05588	05589	05590	05591
12730	05592	05593	05594	05595	05596	05597	05598	05599
12740	05600	05601	05602	05603	05604	05605	05606	05607
12750	05608	05609	05610	05611	05612	05613	05614	05615
12760	05616	05617	05618	05619	05620	05621	05622	05623
12770	05624	05625	05626	05627	05628	05629	05630	05631

OCTAL	0	1	2	3	4	5	6	7
13000	05632	05633	05634	05635	05636	05637	05638	05639
13010	05640	05641	05642	05643	05644	05645	05646	05647
13020	05648	05649	05650	05651	05652	05653	05654	05655
13030	05656	05657	05658	05659	05660	05661	05662	05663
13040	05664	05665	05666	05667	05668	05669	05670	05671
13050	05672	05673	05674	05675	05676	05677	05678	05679
13060	05680	05681	05682	05683	05684	05685	05686	05687
13070	05688	05689	05690	05691	05692	05693	05694	05695
13100	05696	05697	05698	05699	05700	05701	05702	05703
13110	05704	05705	05706	05707	05708	05709	05710	05711
13120	05712	05713	05714	05715	05716	05717	05718	05719
13130	05720	05721	05722	05723	05724	05725	05726	05727
13140	05728	05729	05730	05731	05732	05733	05734	05735
13150	05736	05737	05738	05739	05740	05741	05742	05743
13160	05744	05745	05746	05747	05748	05749	05750	05751
13170	05752	05753	05754	05755	05756	05757	05758	05759
13200	05760	05761	05762	05763	05764	05765	05766	05767
13210	05768	05769	05770	05771	05772	05773	05774	05775
13220	05776	05777	05778	05779	05780	05781	05782	05783
13230	05784	05785	05786	05787	05788	05789	05790	05791
13240	05792	05793	05794	05795	05796	05797	05798	05799
13250	05800	05801	05802	05803	05804	05805	05806	05807
13260	05808	05809	05810	05811	05812	05813	05814	05815
13270	05816	05817	05818	05819	05820	05821	05822	05823
13300	05824	05825	05826	05827	05828	05829	05830	05831
13310	05832	05833	05834	05835	05836	05837	05838	05839
13320	05840	05841	05842	05843	05844	05845	05846	05847
13330	05848	05849	05850	05851	05852	05853	05854	05855
13340	05856	05857	05858	05859	05860	05861	05862	05863
13350	05864	05865	05866	05867	05868	05869	05870	05871
13360	05872	05873	05874	05875	05876	05877	05878	05879
13370	05880	05881	05882	05883	05884	05885	05886	05887
13400	05888	05889	05890	05891	05892	05893	05894	05895
13410	05896	05897	05898	05899	05900	05901	05902	05903
13420	05904	05905	05906	05907	05908	05909	05910	05911
13430	05912	05913	05914	05915	05916	05917	05918	05919
13440	05920	05921	05922	05923	05924	05925	05926	05927
13450	05928	05929	05930	05931	05932	05933	05934	05935
13460	05936	05937	05938	05939	05940	05941	05942	05943
13470	05944	05945	05946	05947	05948	05949	05950	05951
13500	05952	05953	05954	05955	05956	05957	05958	05959
13510	05960	05961	05962	05963	05964	05965	05966	05967
13520	05968	05969	05970	05971	05972	05973	05974	05975
13530	05976	05977	05978	05979	05980	05981	05982	05983
13540	05984	05985	05986	05987	05988	05989	05990	05991
13550	05992	05993	05994	05995	05996	05997	05998	05999
13560	06000	06001	06002	06003	06004	06005	06006	06007
13570	06008	06009	06010	06011	06012	06013	06014	06015
13600	06016	06017	06018	06019	06020	06021	06022	06023
13610	06024	06025	06026	06027	06028	06029	06030	06031
13620	06032	06033	06034	06035	06036	06037	06038	06039
13630	06040	06041	06042	06043	06044	06045	06046	06047
13640	06048	06049	06050	06051	06052	06053	06054	06055
13650	06056	06057	06058	06059	06060	06061	06062	06063
13660	06064	06065	06066	06067	06068	06069	06070	06071
13670	06072	06073	06074	06075	06076	06077	06078	06079
13700	06080	06081	06082	06083	06084	06085	06086	06087
13710	06088	06089	06090	06091	06092	06093	06094	06095
13720	06096	06097	06098	06099	06100	06101	06102	06103
13730	06104	06105	06106	06107	06108	06109	06110	06111
13740	06112	06113	06114	06115	06116	06117	06118	06119
13750	06120	06121	06122	06123	06124	06125	06126	06127
13760	06128	06129	06130	06131	06132	06133	06134	06135
13770	06136	06137	06138	06139	06140	06141	06142	06143

OCTAL	0	1	2	3	4	5	6	7
14000	06144	06145	06146	06147	06148	06149	06150	06151
14010	06152	06153	06154	06155	06156	06157	06158	06159
14020	06160	06161	06162	06163	06164	06165	06166	06167
14030	06168	06169	06170	06171	06172	06173	06174	06175
14040	06176	06177	06178	06179	06180	06181	06182	06183
14050	06184	06185	06186	06187	06188	06189	06190	06191
14060	06192	06193	06194	06195	06196	06197	06198	06199
14070	06200	06201	06202	06203	06204	06205	06206	06207
14100	06208	06209	06210	06211	06212	06213	06214	06215
14110	06216	06217	06218	06219	06220	06221	06222	06223
14120	06224	06225	06226	06227	06228	06229	06230	06231
14130	06232	06233	06234	06235	06236	06237	06238	06239
14140	06240	06241	06242	06243	06244	06245	06246	06247
14150	06248	06249	06250	06251	06252	06253	06254	06255
14160	06256	06257	06258	06259	06260	06261	06262	06263
14170	06264	06265	06266	06267	06268	06269	06270	06271
14200	06272	06273	06274	06275	06276	06277	06278	06279
14210	06280	06281	06282	06283	06284	06285	06286	06287
14220	06288	06289	06290	06291	06292	06293	06294	06295
14230	06296	06297	06298	06299	06300	06301	06302	06303
14240	06304	06305	06306	06307	06308	06309	06310	06311
14250	06312	06313	06314	06315	06316	06317	06318	06319
14260	06320	06321	06322	06323	06324	06325	06326	06327
14270	06328	06329	06330	06331	06332	06333	06334	06335
14300	06336	06337	06338	06339	06340	06341	06342	06343
14310	06344	06345	06346	06347	06348	06349	06350	06351
14320	06352	06353	06354	06355	06356	06357	06358	06359
14330	06360	06361	06362	06363	06364	06365	06366	06367
14340	06368	06369	06370	06371	06372	06373	06374	06375
14350	06376	06377	06378	06379	06380	06381	06382	06383
14360	06384	06385	06386	06387	06388	06389	06390	06391
14370	06392	06393	06394	06395	06396	06397	06398	06399
14400	06400	06401	06402	06403	06404	06405	06406	06407
14410	06408	06409	06410	06411	06412	06413	06414	06415
14420	06416	06417	06418	06419	06420	06421	06422	06423
14430	06424	06425	06426	06427	06428	06429	06430	06431
14440	06432	06433	06434	06435	06436	06437	06438	06439
14450	06440	06441	06442	06443	06444	06445	06446	06447
14460	06448	06449	06450	06451	06452	06453	06454	06455
14470	06456	06457	06458	06459	06460	06461	06462	06463
14500	06464	06465	06466	06467	06468	06469	06470	06471
14510	06472	06473	06474	06475	06476	06477	06478	06479
14520	06480	06481	06482	06483	06484	06485	06486	06487
14530	06488	06489	06490	06491	06492	06493	06494	06495
14540	06496	06497	06498	06499	06500	06501	06502	06503
14550	06504	06505	06506	06507	06508	06509	06510	06511
14560	06512	06513	06514	06515	06516	06517	06518	06519
14570	06520	06521	06522	06523	06524	06525	06526	06527
14600	06528	06529	06530	06531	06532	06533	06534	06535
14610	06536	06537	06538	06539	06540	06541	06542	06543
14620	06544	06545	06546	06547	06548	06549	06550	06551
14630	06552	06553	06554	06555	06556	06557	06558	06559
14640	06560	06561	06562	06563	06564	06565	06566	06567
14650	06568	06569	06570	06571	06572	06573	06574	06575
14660	06576	06577	06578	06579	06580	06581	06582	06583
14670	06584	06585	06586	06587	06588	06589	06590	06591
14700	06592	06593	06594	06595	06596	06597	06598	06599
14710	06600	06601	06602	06603	06604	06605	06606	06607
14720	06608	06609	06610	06611	06612	06613	06614	06615
14730	06616	06617	06618	06619	06620	06621	06622	06623
14740	06624	06625	06626	06627	06628	06629	06630	06631
14750	06632	06633	06634	06635	06636	06637	06638	06639
14760	06640	06641	06642	06643	06644	06645	06646	06647
14770	06648	06649	06650	06651	06652	06653	06654	06655

OCTAL	0	1	2	3	4	5	6	7
15000	06656	06657	06658	06659	06660	06661	06662	06663
15010	06664	06665	06666	06667	06668	06669	06670	06671
15020	06672	06673	06674	06675	06676	06677	06678	06679
15030	06680	06681	06682	06683	06684	06685	06686	06687
15040	06688	06689	06690	06691	06692	06693	06694	06695
15050	06696	06697	06698	06699	06700	06701	06702	06703
15060	06704	06705	06706	06707	06708	06709	06710	06711
15070	06712	06713	06714	06715	06716	06717	06718	06719
15100	06720	06721	06722	06723	06724	06725	06726	06727
15110	06728	06729	06730	06731	06732	06733	06734	06735
15120	06736	06737	06738	06739	06740	06741	06742	06743
15130	06744	06745	06746	06747	06748	06749	06750	06751
15140	06752	06753	06754	06755	06756	06757	06758	06759
15150	06760	06761	06762	06763	06764	06765	06766	06767
15160	06768	06769	06770	06771	06772	06773	06774	06775
15170	06776	06777	06778	06779	06780	06781	06782	06783
15200	06784	06785	06786	06787	06788	06789	06790	06791
15210	06792	06793	06794	06795	06796	06797	06798	06799
15220	06800	06801	06802	06803	06804	06805	06806	06807
15230	06808	06809	06810	06811	06812	06813	06814	06815
15240	06816	06817	06818	06819	06820	06821	06822	06823
15250	06824	06825	06826	06827	06828	06829	06830	06831
15260	06832	06833	06834	06835	06836	06837	06838	06839
15270	06840	06841	06842	06843	06844	06845	06846	06847
15300	06848	06849	06850	06851	06852	06853	06854	06855
15310	06856	06857	06858	06859	06860	06861	06862	06863
15320	06864	06865	06866	06867	06868	06869	06870	06871
15330	06872	06873	06874	06875	06876	06877	06878	06879
15340	06880	06881	06882	06883	06884	06885	06886	06887
15350	06888	06889	06890	06891	06892	06893	06894	06895
15360	06896	06897	06898	06899	06900	06901	06902	06903
15370	06904	06905	06906	06907	06908	06909	06910	06911
15400	06912	06913	06914	06915	06916	06917	06918	06919
15410	06920	06921	06922	06923	06924	06925	06926	06927
15420	06928	06929	06930	06931	06932	06933	06934	06935
15430	06936	06937	06938	06939	06940	06941	06942	06943
15440	06944	06945	06946	06947	06948	06949	06950	06951
15450	06952	06953	06954	06955	06956	06957	06958	06959
15460	06960	06961	06962	06963	06964	06965	06966	06967
15470	06968	06969	06970	06971	06972	06973	06974	06975
15500	06976	06977	06978	06979	06980	06981	06982	06983
15510	06984	06985	06986	06987	06988	06989	06990	06991
15520	06992	06993	06994	06995	06996	06997	06998	06999
15530	07000	07001	07002	07003	07004	07005	07006	07007
15540	07008	07009	07010	07011	07012	07013	07014	07015
15550	07016	07017	07018	07019	07020	07021	07022	07023
15560	07024	07025	07026	07027	07028	07029	07030	07031
15570	07032	07033	07034	07035	07036	07037	07038	07039
15600	07040	07041	07042	07043	07044	07045	07046	07047
15610	07048	07049	07050	07051	07052	07053	07054	07055
15620	07056	07057	07058	07059	07060	07061	07062	07063
15630	07064	07065	07066	07067	07068	07069	07070	07071
15640	07072	07073	07074	07075	07076	07077	07078	07079
15650	07080	07081	07082	07083	07084	07085	07086	07087
15660	07088	07089	07090	07091	07092	07093	07094	07095
15670	07096	07097	07098	07099	07100	07101	07102	07103
15700	07104	07105	07106	07107	07108	07109	07110	07111
15710	07112	07113	07114	07115	07116	07117	07118	07119
15720	07120	07121	07122	07123	07124	07125	07126	07127
15730	07128	07129	07130	07131	07132	07133	07134	07135
15740	07136	07137	07138	07139	07140	07141	07142	07143
15750	07144	07145	07146	07147	07148	07149	07150	07151
15760	07152	07153	07154	07155	07156	07157	07158	07159
15770	07160	07161	07162	07163	07164	07165	07166	07167

OCTAL	0	1	2	3	4	5	6	7
16000	07168	07169	07170	07171	07172	07173	07174	07175
16010	07176	07177	07178	07179	07180	07181	07182	07183
16020	07184	07185	07186	07187	07188	07189	07190	07191
16030	07192	07193	07194	07195	07196	07197	07198	07199
16040	07200	07201	07202	07203	07204	07205	07206	07207
16050	07208	07209	07210	07211	07212	07213	07214	07215
16060	07216	07217	07218	07219	07220	07221	07222	07223
16070	07224	07225	07226	07227	07228	07229	07230	07231
16100	07232	07233	07234	07235	07236	07237	07238	07239
16110	07240	07241	07242	07243	07244	07245	07246	07247
16120	07248	07249	07250	07251	07252	07253	07254	07255
16130	07256	07257	07258	07259	07260	07261	07262	07263
16140	07264	07265	07266	07267	07268	07269	07270	07271
16150	07272	07273	07274	07275	07276	07277	07278	07279
16160	07280	07281	07282	07283	07284	07285	07286	07287
16170	07288	07289	07290	07291	07292	07293	07294	07295
16200	07296	07297	07298	07299	07300	07301	07302	07303
16210	07304	07305	07306	07307	07308	07309	07310	07311
16220	07312	07313	07314	07315	07316	07317	07318	07319
16230	07320	07321	07322	07323	07324	07325	07326	07327
16240	07328	07329	07330	07331	07332	07333	07334	07335
16250	07336	07337	07338	07339	07340	07341	07342	07343
16260	07344	07345	07346	07347	07348	07349	07350	07351
16270	07352	07353	07354	07355	07356	07357	07358	07359
16300	07360	07361	07362	07363	07364	07365	07366	07367
16310	07368	07369	07370	07371	07372	07373	07374	07375
16320	07376	07377	07378	07379	07380	07381	07382	07383
16330	07384	07385	07386	07387	07388	07389	07390	07391
16340	07392	07393	07394	07395	07396	07397	07398	07399
16350	07400	07401	07402	07403	07404	07405	07406	07407
16360	07408	07409	07410	07411	07412	07413	07414	07415
16370	07416	07417	07418	07419	07420	07421	07422	07423
16400	07424	07425	07426	07427	07428	07429	07430	07431
16410	07432	07433	07434	07435	07436	07437	07438	07439
16420	07440	07441	07442	07443	07444	07445	07446	07447
16430	07448	07449	07450	07451	07452	07453	07454	07455
16440	07456	07457	07458	07459	07460	07461	07462	07463
16450	07464	07465	07466	07467	07468	07469	07470	07471
16460	07472	07473	07474	07475	07476	07477	07478	07479
16470	07480	07481	07482	07483	07484	07485	07486	07487
16500	07488	07489	07490	07491	07492	07493	07494	07495
16510	07496	07497	07498	07499	07500	07501	07502	07503
16520	07504	07505	07506	07507	07508	07509	07510	07511
16530	07512	07513	07514	07515	07516	07517	07518	07519
16540	07520	07521	07522	07523	07524	07525	07526	07527
16550	07528	07529	07530	07531	07532	07533	07534	07535
16560	07536	07537	07538	07539	07540	07541	07542	07543
16570	07544	07545	07546	07547	07548	07549	07550	07551
16600	07552	07553	07554	07555	07556	07557	07558	07559
16610	07560	07561	07562	07563	07564	07565	07566	07567
16620	07568	07569	07570	07571	07572	07573	07574	07575
16630	07576	07577	07578	07579	07580	07581	07582	07583
16640	07584	07585	07586	07587	07588	07589	07590	07591
16650	07592	07593	07594	07595	07596	07597	07598	07599
16660	07600	07601	07602	07603	07604	07605	07606	07607
16670	07608	07609	07610	07611	07612	07613	07614	07615
16700	07616	07617	07618	07619	07620	07621	07622	07623
16710	07624	07625	07626	07627	07628	07629	07630	07631
16720	07632	07633	07634	07635	07636	07637	07638	07639
16730	07640	07641	07642	07643	07644	07645	07646	07647
16740	07648	07649	07650	07651	07652	07653	07654	07655
16750	07656	07657	07658	07659	07660	07661	07662	07663
16760	07664	07665	07666	07667	07668	07669	07670	07671
16770	07672	07673	07674	07675	07676	07677	07678	07679