ALGORITHMS PART 2

INTRODUCTION TO COMPUTER ENGINEERING

BLM1551

ASIST. PROF. DR. Z. CIHAN TAYŞİ

OUTLINE

- Algorithms
  - Variable concept, Loops, Array concept, Example problems
- Number systems
  - Binary, octal, decimal, hexadecimal
- Operators
  - Arithmetic, logical, bitwise

ALGORITHMS

VARIABLE CONCEPT, LOOPS, ARRAY CONCEPT, EXAMPLE PROBLEMS

WHAT IS A VARIABLE ?

- **Variables** are used to store information to be referenced and manipulated in a computer program.
- They also provide a way of labeling data with a descriptive name, so our programs can be understood more clearly by the reader and ourselves.
- It is helpful to think of variables as containers that hold information. Their sole purpose is to label and store data in memory. This data can then be used throughout your program.

EXAMPLE #1

- Please find the minimum of the given three numbers...

- We will receive 3 numbers from the user
- We will find the minimum among those numbers

EXAMPLE #2

- Please find the properties of a given triangle

- We will receive dimensions (3 edges) of a triangle
- We will state the properties of the triangle
  - if all edges are same => EŞ KENAR
  - if just two edges are equal => İKİZ KENAR
  - if none of the edges are equal => ÇEŞİT KENAR

EXAMPLE #3

- Develop an algorithm that uses minimum number of bills to pay a given amount of money.

- user will supply the amount of money
- we already know the available bills (200, 100, 50, 20, 10, 5)

LOOPS

- Deterministic loops
  - The number of iterations of such a loop are known in advance, even before the loop has started.
  - Most counting loops are deterministic.
  - Before they start, we can say how many times they will execute
- non-Deterministic loops
  - A loop that is driven by the response of a user is not deterministic, because we cannot predict the response of the user.
  - Non-deterministic loops usually are controlled by a Boolean,
  - and the number of iterations is not known in advance

A RANGE OF NUMBERS

- find the average of given numbers

- First, user will tell us number (N) of elements it will give
- Then he/she will supply the numbers one by one
- We should add up numbers and after the last element divide the summation by N

FIBONACCI NUMBERS

- Please find the N^{th} Fibonacci number

- Fibonacci numbers
  - $F_{i+1} = F_i + F_{i-1}$
  - starts from 1 and goes like 1, 1, 2, 3, 5, 8, 13, 21

NEWTONS SQUARE ROOT

- We will calculate the square root of a given number

- user will supply a number
- we also need a threshold value
  - preferably supplied by the user
- you know the formula !
  - calculate a new estimation using the formula
  - iterate till the difference between iterations is smaller then the threshold

ARRAYS

- An array is a group of related data values (called elements) that are grouped together.
- **In most cases,** all of the array elements must be the **same data type**.

ORGANIZING AN ARRAY

- Please reverse the position of elements in a given array

- user will supply number of elements
- user will supply elements of the array
- we will switch first element with the last and continue to do it till all elements are switched

ORGANIZING AN ARRAY

- Please develop an algorithm that positions odd elements in the beginning and the even elements at the end of a given array.

- user will supply number of elements
- user will supply elements of the array
- we will examine each element and relocate it accordingly, if necessary

ORGANIZING AN ARRAY

- Please develop an algorithm that finds the position of the element with the minimum value

- user will supply number of elements
- user will supply elements of the array
- we will examine each element and decide if it is minimum
- if yes, we will store its position

NUMBER SYSTEMS

BINARY, OCTAL, DECIMAL, HEXADECIMAL

DECIMAL SYSTEM

- Decimal number system, also called Hindu-Arabic, or Arabic, number system, in mathematics, positional numeral system employing 10 as the base and requiring 10 different numerals, the digits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.
- It also **requires a dot (decimal point) to represent decimal fractions**.
- In this scheme, the numerals used in denoting a number take different place values depending upon position.
- In a base-10 system the number 543.21 represents the sum $(5 \times 10^2) + (4 \times 10^1) + (3 \times 10^0) + (2 \times 10^{-1}) + (1 \times 10^{-2})$.

BINARY SYSTEM

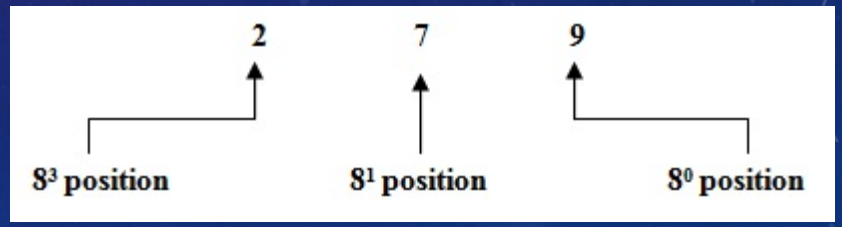
- In the binary system, in which the base is 2, there are just two digits, 0 and 1; the number two must be represented here as 10, since it plays the same role as does ten in the decimal system.
- A binary number is generally much longer than its corresponding decimal number;
  - for example, 256,058 has the binary representation 111 11010 00001 11010.
- The reason for the greater length of the binary number is that a binary digit distinguishes between only two possibilities, 0 or 1, whereas a decimal digit distinguishes among 10 possibilities; in other words, a binary digit carries less information than a decimal digit.
  - Because of this, **its name has been shortened to bit**;

OCTAL SYSTEM

- Octal number system uses only 8 numbers to represent the numbers, so it has the name **"Octal".**
  - They are 0, 1, 2, 3, 4, 5, 6 and 7

- In octal numbering system we represent the binary numbers as a set of 3 digits (2^3= 8).

- Octal has been used of the 12-bit, 24-bit, and 36-bit processors.
  - E.g. PDP-8, ICL 1900 and IBM mainframes



HEXADECIMAL SYSTEM

- Hexadecimal Number System (Base-16)
- The hexadecimal number system uses SIXTEEN values to represent numbers.
- The values are, **0 1 2 3 4 5 6 7 8 9 A B C D E F** with 0 having the least value and F having the greatest value.
- Columns are used in the same way as in the decimal system, in that the left most column is used to represent the greatest value.
- Example :
- $(256{,}058)_{10}$ $(11\ 1110\ 1000\ 0011\ 1010)_{2}$ $(764072)_{8}$ $(3E83A)_{16}$

OPERATORS

ARITHMETIC, LOGICAL, BITWISE

ARITHMETIC OPERATORS

- addition (+)
- subtraction (-)
- multiplication (*)
- division (/)
- remainder (mod)

RELATIONAL OPERATORS

- greater than (>)
- less than (<)
- greater than or equal to (>=)
- less than or equal to (<=)
- equal to (=)
- not equal to (≠)

LOGIC OPERATORS

- logical AND (AND)
- logical OR (OR)
- logical negation (NOT)

BITWISE OPERATORS

- AND operator (&)
- OR operator (|)
- Exclusive OR (XOR) operator (XOR)
- Shift operator (<<) (>>)
  - left / right
- Rotate operation
  - left / right

AND OPERATOR

- **AND operator copies a bit to the result if it exists in both operands.**
- Assume A = 60 and B = 13 in binary format, they will be as follows −
- A = 0011 1100
- B = 0000 1101

- A&B = ?
- **0000 1100**

A	B	X= A AND B
0	0	0
0	1	0
1	0	0
1	1	1

OR OPERATOR

- **OR operator copies a bit if it exists in either operand.**
- Assume A = 60 and B = 13 in binary format, they will be as follows −
- A = 0011 1100
- B = 0000 1101

- A | B = ?
- **0011 1101**

A	B	X= A OR B
0	0	0
0	1	1
1	0	1
1	1	1

XOR OPERATOR

- **XOR Operator copies the bit if it is set in one operand but not both**.
- Assume A = 60 and B = 13 in binary format, they will be as follows −
- A = 0011 1100
- B = 0000 1101

- A XOR B = ?
- **0011 0001**

Inputs		Output
A	*B*	*X*
0	0	1
0	1	0
1	0	0
1	1	1

SHIFT OPERATOR

- **Remember A is 60!**

- Left Shift Operator.
  - The left operands value is moved left by the number of bits specified by the right operand.
  - A << 2 = 240 (1111 0000)
- Binary Right Shift Operator
  - The left operands value is moved right by the number of bits specified by the right operand.
  - A >> 2 = 15 (0000 1111)

ROTATE OPERATION