

Module-1

PROBLEM SOLVING USING C PROGRAMMING

Problem Solving techniques: Introduction, Problem solving procedure, Algorithm: Steps involved in algorithm development. **Algorithms for simple problems:** To find largest of three numbers, factorial of number, check for prime number, check for palindrome, Count no. of odd, even and zeros in list of integers.

Flowcharts: Definition, advantages, Symbols used in flow charts. Flowcharts for simple problems mentioned in algorithms. Pseudo code.

Introduction

A computer is a very powerful and versatile machine capable of performing a multitude of different tasks, yet it has no intelligence or thinking power. The intelligence Quotient (I.Q) of a computer is zero. A computer performs many tasks exactly in the same manner as it is told to do. This places responsibility on the user to instruct the computer in a correct and precise manner, so that the machine is able to perform the required job in a proper way. A wrong or ambiguous instruction may sometimes prove disastrous. In order to instruct a computer correctly, the user must have clear understanding of the problem to be solved. A part from this he should be able to develop a method, in the form of series of sequential steps, to solve it. Once the problem is well-defined and a method of solving it is developed, then instructing the computer to solve the problem becomes relatively easier task. Thus, before attempt to write a computer program to solve a given problem. It is necessary to formulate or define the problem in a precise manner. Once the problem is defined, the steps required to solve it, must be stated clearly in the required order.

Procedure (Steps Involved in Problem Solving)

A computer cannot solve a problem on its own. One has to provide step by step solutions of the problem to the computer. In fact, the task of problem solving is not that of the computer. It is the programmer who has to write down the solution to the problem in terms of simple operations which the computer can understand and execute. In order to solve a problem by the computer, one has to pass through certain stages or steps. They are

1. Problem Definition
2. Problem Analysis
3. Algorithm Design
4. Coding
5. Testing & Debugging
6. Maintenance

1. Problem Definition

In this phase, we define the problem statement and we decide the boundaries of the problem. we need to understand the problem statement, what is our requirement, what should be the output of the problem statement. These are defined in this first phase of the program development life cycle.

Module-1

PROBLEM SOLVING USING C PROGRAMMING

2. Problem Analysis

In phase 2, we determine the requirements like variables, functions, etc. to solve the problem. That means we gather the required resources to solve the problem defined in the problem definition phase. We also determine the bounds of the solution.

3. Algorithm Design

During this phase, we develop a step by step procedure to solve the problem using the specification given in the previous phase. This phase is very important for program development. That means we write the solution in step by step statements.

4. Coding

This phase uses a programming language to write or implement actual programming instructions for the steps defined in the previous phase. we construct actual program. That means we write the program to solve the given problem using programming languages like C, C++, Java etc.,

5. Testing & Debugging

Testing involves verification of the correctness of the program created. Testing is normally done by running the program with all types of sample data and then observing the output.

Debugging is the process of detection and correction of errors in the program code like syntax errors, runtime errors and logical errors.

1. Maintenance

Maintenance of program is another vital step. Due to the fact that the user requirement keep changing, the programs also need to be changed to meet the changing requirement.

Algorithms

The typical meaning of an algorithm is a formally defined procedure for performing some calculation. If a procedure is formally defined, then it must be implemented using some formal language, and such languages are known as programming languages. The algorithm gives the logic of the program, that is, a step-by-step description of how to arrive at a solution. In general terms, an algorithm provides a blueprint to writing a program to solve a particular problem. It is considered to be an effective procedure for solving a problem in a finite number of steps. That is, a well-defined algorithm always provides an answer, and is guaranteed to terminate. Algorithms are mainly used to achieve software reuse. Once we have an idea or a blueprint of a solution, we can implement it in any high-level language, such as C, C++, Java, and so on.

An algorithm is a step by step procedure to solve a given problem in finite number of steps.

Properties of an Algorithm

Module-1

PROBLEM SOLVING USING C PROGRAMMING

Algorithms consist of statement written in simple English and mathematical expressions.

Every algorithm must exhibit the following properties :

1. It must be simple.
2. It must be precise with no ambiguity.
3. It must accept zero or more inputs.
4. It must give a unique solution to the problem.
5. It must involve a finite number of instructions.

Algorithmic Notations

An algorithm is written using the following algorithmic notations.

- i) **Name** : Specifies the problem.
- ii) **Step number** : Each instruction is identified by a number called the step number.
- iii) **Comments** : It describes the operation and is written within a pair of square brackets[].
- iv) **Beginning/Termination** : It indicates the beginning/end of the algorithm with a START/STOP statement

Advantages of Algorithms

1. It is a step by step solution which is easily understood since it is written using English like language.
2. Algorithm consists of finite number of steps to arrive at a solution.
3. Debugging is easy, since the logical sequence is listed stepwise.
4. It is independent of any programming language so that the user need not know the syntax of a programming language to develop the logic.
5. Algorithm makes program maintenance easier.

Disadvantages of Algorithms

1. Developing algorithms for large and complex problems would be time consuming and difficult to understand.
2. Understanding complex logic through algorithms would be difficult.

Examples:

1. **Problem definition** : Develop an algorithm to find the largest of three numbers.

Problem Analysis :

input - a,b,c

output – string

Module-1

PROBLEM SOLVING USING C PROGRAMMING

Algorithm

step 1: start

step 2: input a,b,c

Step 3: if (a>b) and (a>c) then print "a is greater" _

Else if (b>a) and (b>c) then print "b is greater"

Step 4: stop

2. Problem definition: : Develop an algorithm to find the factorial of number.

Problem Analysis :

input - n

output – string

Algorithm

step 1. Start

step 2. Read the number n

step 3. [Initialize]

i=1, fact=1

step 4. Repeat step 4 through 6 until i<=n

step 5. fact=fact*i

step 6. i=i+1

step 7. Print fact

step 8. Stop

3. Problem definition: Develop an algorithm to find check weather palindrome or not.

Problem Analysis :

input - n

output – string

Algorithm

step 1. Start

step 2. Accept n

step 3. [Initialize] rev=0, a=n

step 4. Repeat steps 4 to 7 until n!=0

step 5. rem=n%10

step 6. rev=rev*10+rem

step 7. n=n/10

step 8. if rev==a

step 9. print "It is a palindrome number"

Module-1

PROBLEM SOLVING USING C PROGRAMMING

step 10. else print "It is not a palindrome number"

step 11. Stop

- 4. Problem definition: Develop an algorithm to find Count no. of even, odd and zeros in list of integers.**

Problem Analysis :

input - n

output – string

Algorithm

step 1. Start

step 2. Accept n

step 3. [Initialize] even_count=0, odd_count=0 and zero_count=0;

step 4. if(a[i]==0)

step 5. zero++

step 6. else if(a[i]%2==0)

step 7. even++

step 8. else if(a[i]%2!=0)

step 9. odd++

step 10. display even,odd,zero

step 11. Stop

- 5. Develop an algorithm to find the average of three numbers taken as input from the user.**

step 1. Start

step 2. Input a,b,c

step 3. Compute sum=a+b+c

step 4. Avg=sum/3

step 5. Display avg

step 6. End

- 6. Develop an algorithm to find the maximum of two numbers taken as input from the user.**

step 1. Start

step 2. Input a,b

step 3. if(a>b)

step 4. max=a

else

max=b

step 5. Display max

step 6. stop

Module-1

PROBLEM SOLVING USING C PROGRAMMING

Flowcharts

A flowchart is a pictorial representation of an algorithm. It specifies the solution procedure, the relevant computations, points of decisions, and other information required for the solution. Flowcharts are created by using special geometric symbols. Each symbol represents an operation.

A flowchart is a graphical or symbolic representation of a process. It is basically used to design and document virtually complex processes to help the viewers to visualize the logic of the process, so that they can gain a better understanding of the process and find flaws, bottlenecks and other less obvious features within it.

The symbols are joined by arrows to obtain a complete solution.

Advantages of flowcharts

- Flowcharts form a good visual aid to represent the logic for the problem solution.
- It is a form of program documentation.
- With the help of the flowchart, program could be coded efficiently.
- It facilitates orderly debugging and testing of programs.
- It provides efficient program maintenance in that the programmer can concentrate only on that part which is to be modified.

Rules for writing flowcharts

- Flowcharts are drawn from top to bottom or left to right.
- A flowchart always begins with START symbol and ends with STOP symbol.
- Flow lines are used to join the symbols.
- There should be at least one STOP symbol in any flow chart

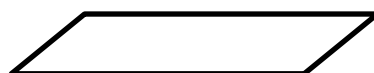
Symbols used in Flow-Charts

The symbols that we make use while drawing flowcharts as given below are as per conventions followed by International Standard Organization (ISO).

- a. **Oval:** Rectangle with rounded sides is used to indicate either START/ STOP of the program. ..



- b. **Input and output indicators:** Parallelograms are used to represent input and output operations. Statements like INPUT, READ and PRINT are represented in these Parallelograms.



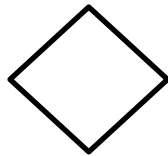
Module-1

PROBLEM SOLVING USING C PROGRAMMING

- c. **Process Indicators:** - Rectangle is used to indicate any set of processing operation such as for storing arithmetic operations.



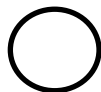
- d. **Decision Makers:** The diamond is used for indicating the step of decision making and therefore known as decision box. Decision boxes are used to test the conditions or ask questions and depending upon the answers, the appropriate actions are taken by the computer.



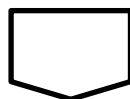
- e. **Flow lines** indicate the direction being followed in the flowchart. In a Flowchart, every line must have an arrow on it to indicate the direction. The arrows may be in any direction.



- f. **On- Page connectors:** Circles are used to join the different parts of a flowchart and these circles are called on-page connectors. The uses of these connectors give a neat shape to the flowcharts. In a complicated problems, a flowchart may run in to several pages. The parts of the flowchart on different pages are to be joined with each other. The parts to be joined are indicated by the circle.




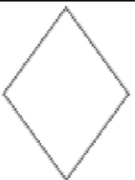



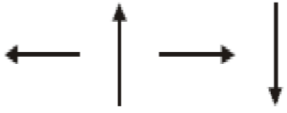


- g. **Off-page connectors:** This connector represents a break in the path of flowchart which is too large to fit on a single page. It is similar to on-page connector. The connector symbol marks where the algorithm ends on the first page and where it continues on the second.



Module-1

PROBLEM SOLVING USING C PROGRAMMING

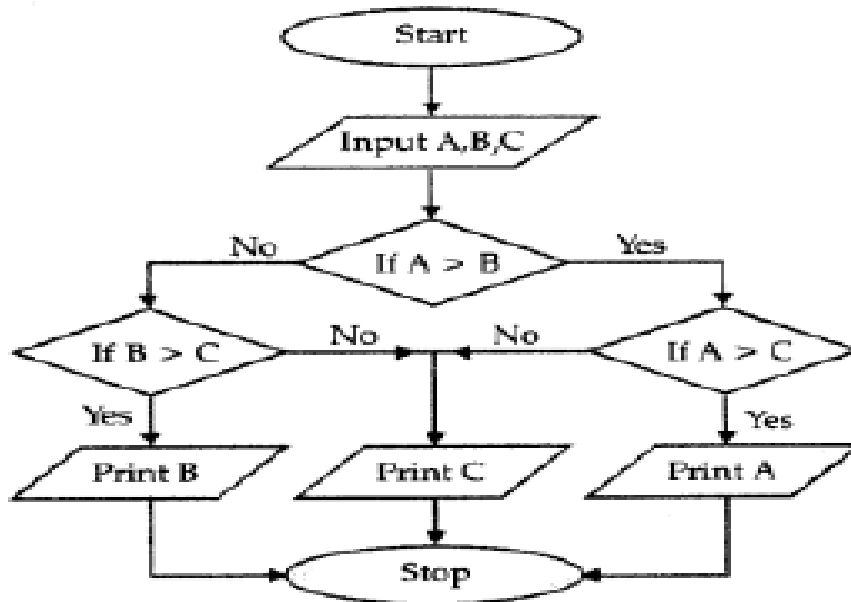
Symbol	Meaning
	Terminal (Start/Stop) Indicates the beginning or end of a program.
	Input/Output Used to read or print data or information.
	Processing Represents Calculations, Processing or data manipulation.
	Decision Represents comparisons or decisions and branching
	Looping (Repetition) Represent a group of instructions to be executed repeatedly.
	Pre-defined Process (Module/Subprogram) Represents pre-defined computations i.e., a group of operations specified elsewhere.
	Connector Indicates an entry or an exit to another part of the flowchart.
	Director of flow Indicates the direction of processing or flow of control.

Module-1

PROBLEM SOLVING USING C PROGRAMMING

Example:

1. Largest of three numbers



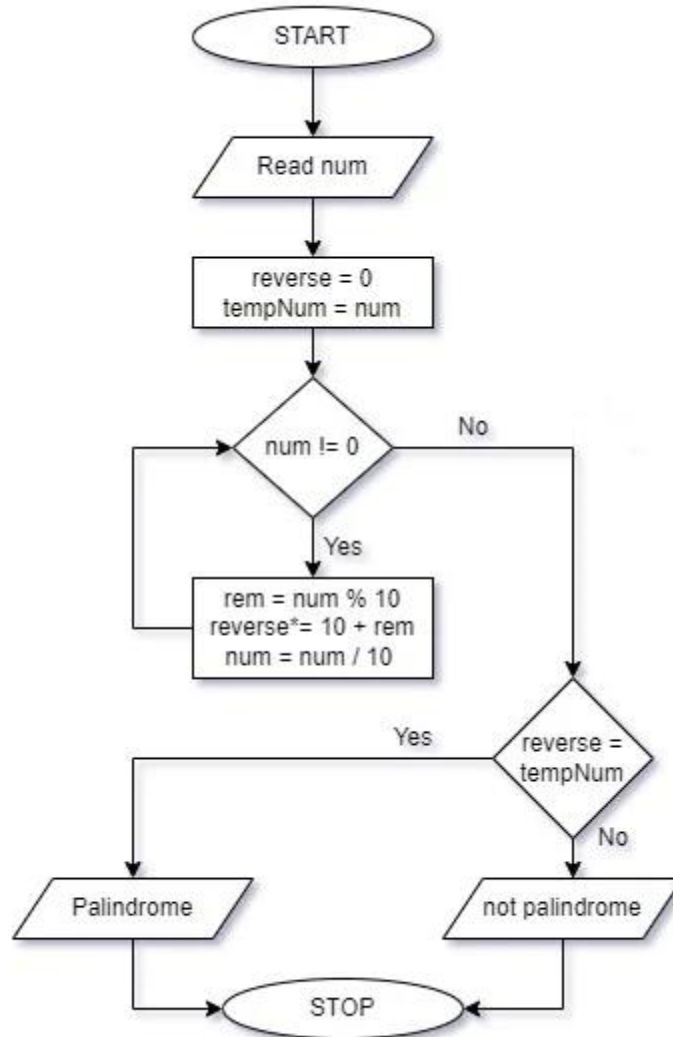
2. Factorial of a number



Module-1

PROBLEM SOLVING USING C PROGRAMMING

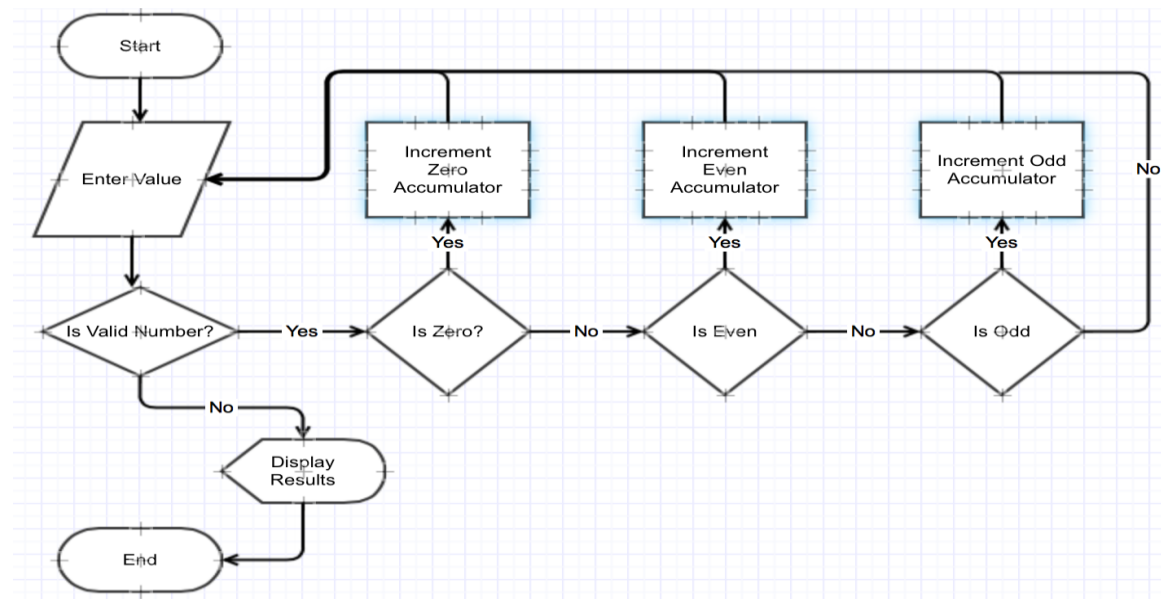
3. Palindrome or not



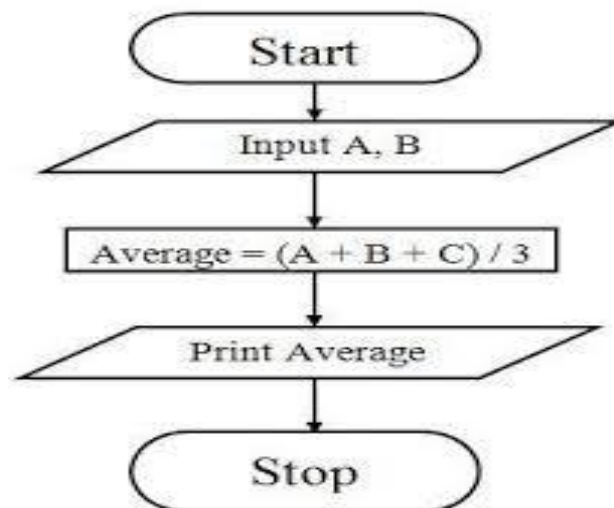
Module-1

PROBLEM SOLVING USING C PROGRAMMING

4. Count no of even, odd and zeros in a list of integers.



5. Average of three numbers



Module-1

PROBLEM SOLVING USING C PROGRAMMING

Pseudo code

The Pseudo code is neither an algorithm nor a program. It is an abstract form of a program. It consists of English like statements which perform the specific operations. It is defined for an algorithm. It does not use any graphical representation. In pseudo code, the program is represented in terms of words and phrases, but the syntax of program is not strictly followed.

Pseudocode is a compact and informal high-level description of an algorithm that uses the structural conventions of a programming language. It facilitates designers to focus on the logic of the algorithm without getting bogged down by the details of language syntax. An ideal pseudocode must be complete, describing the entire logic of the algorithm, so that it can be translated straightaway into a programming language.

Advantages:

- * Easy to read,
- * Easy to understand,
- * Easy to modify.

Example: Write a pseudo code to perform the basic arithmetic operations.

```
Read n1, n2
Sum = n1 + n2
Diff = n1 - n2
Mult = n1 * n2
Quot = n1/n2
Print sum, diff, mult, quot
End
```

Example: Write a pseudo code to perform the Largest of three numbers.

```
START
INPUT num1,num2,num3
IF num1>num2 AND num1>num3
Print num1 is the largest
ELSE IF num2>num1 AND num2>num3
Print num2 is the largest
ELSE
Print num3 is the largest
```

Module-1

PROBLEM SOLVING USING C PROGRAMMING

End

Example: Write a pseudo code to perform Palindrome or not.

Input n

Set temp=n

Set rev=0

Set rem=0

While temp!=0 Loop

Set rem=temp%10

Set rev=rev*10+rem

Set temp=temp/10

EndWhile

If rev=n Then

Output n, "is palindrome"

Else

Output n, "is not palindrome"

EndIf