

Bachelor of Computer Application, Semester–I

Paper-2

C PROGRAMMING



Programming

Presentation by:

Prof. Sachin Y Zade

Assistant Professor,

Kamla Nehru Mahavidyalaya,

Nagpur

Steps of the Day



Let's Start 





Rules of Lecture



Rules



Don't be late to come in my classroom

Rules



Presence must be above 80%

Rules



Do all components of value



Syllabus



Structure of Semester-I

Appendix-A
Teaching & Examination Scheme
Bachelor of Computer Application (B.C.A.)
Three Year (SIX SEMESTER) DEGREE COURSE
B.C.A. Part I (Semester I)

Sr.No.	Papers	Title of Paper	Teaching Scheme			Examination Scheme								
			Th+Tu (Periods)	Pr (Periods)	Total periods	Duration Hours	Theory				Practical			Total (Th,Pr,AD)
Max Marks Th. Papers	Max. Marks IA	Total					Min Passing Marks	Duration Hours	Max Marks Practical	Min Passing Marks				
1.	-	English	4+1	-	4+1	3	60	15	75	30	-	-	-	75
2.	-	Marathi/Hindi/U rdu/Gujarati/ Sanskrit / Suppl. English	3	-	3	3	60	15	75	30	-	-	-	75
3.	Paper-I	Computer Fundamentals	3	-	3	3	50	10	60	24	-	-	-	60
4.	Paper-II	“C” Programming	3	-	3	3	50	10	60	24	-	-	-	60
5.	Paper-III	Statistical Methods	3	-	3	3	50	10	60	24	-	-	-	60
6.	Paper-IV	Discrete Mathematics – I	3	-	3	3	50	10	60	24	-	-	-	60
7.	Paper-V	Operating Systems	3	-	3	3	50	10	60	24	-	-	-	60
8.	Paper-VI	Office Automation	3	-	3	3	50	10	60	24	-	-	-	60
9.	Pactical-I	Practical I – based on paper I & paper II		6	6	-	-	-	-	-	4	30	12	30
10.	Pactical-II	Practical II – based on paper III & paper IV		6	6	-	-	-	-	-	4	30	12	30
11.	Pactical-III	Practical III – based on paper V & paper VI		6	6	-	-	-	-	-	4	30	12	30

Note:

1. Th = Theory; Pr = Practical; Tu = Tutorial; IA = Internal Assessment.
2. Minimum marks for passing will be 40% of the total marks allotted to that paper / practical
3. Candidate has to pass theory papers and practical separately.

Grand Total of Semester I: 450 + 150 = TOTAL = 600 Marks.

- The strength of Batch of Practical and Tutorial for Under Graduates classes shall be 16 with an additional; of 10% with the permission of Hon'ble Vice-Chancellor.
- Details of Course of Languages shall be as per B.Sc. I

B.C.A. Part I Semester I

Paper II

‘C’ PROGRAMMING

UNIT- I :

Programming Structure : Sequence, Selection, Iteration and Modular. **Problem Solving techniques:** Development Tools: Algorithm, Flowcharts and Pseudo code (Definition and its characteristics) **Developing Algorithm and Drawing flowcharts**

UNIT- II :

C Character set, Tokens, Identifier, Keywords, Variables, Data types, Qualifiers. Operators and Expressions: Arithmetic, Relational, Logical, Bit-Wise, Increment, Decrement, Conditional and Special operators. typedef, Type Conversion, Constants, Declaring Symbolic Constants, Character Strings, Enumerated Data Types, Operator Precedence and Associativity. Library functions. : Maths, string handling Functions. Control Structure: Compound Statement, Selection Statement: if, if-else, Nested if, switch. Iteration statement: for, while, do..while, Nested loops, Jump statement: break, continue, goto. (Special emphasis on problem solving)

UNIT- III :

Arrays: Need, Types: Single and Two Dimensional Array.

Strings: Strings Manipulation, Arrays of Strings, Evaluation order

Function: Function Components, Return Data type, Parameter Passing, Return by Reference, Default Arguments, Recursive Functions, Arrays with Functions, Storage Classes. (Special emphasis on problem Solving)

UNIT- IV:

Structure: Declaration, Definition, Accessing structure members, Initialization, Nesting of Structures.

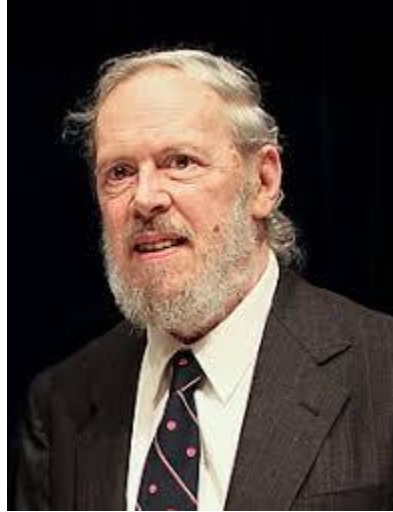
Union: Unions, Differences between Structure and Union

Pointer: Introduction, Address Operator (&), Pointer variables, Void pointers, Pointer Arithmetic, Pointers to Pointers.

File handling: Hierarchy of File Stream Classes, Opening & closing a file, Testing for errors, File Modes, File pointers and their manipulations, Sequential Access, Random Access, Command Line arguments.

Reference Books:

1. The Art of programming through flowcharts & algorithm by Anil B. Chaudhari
Firewall Media, Laxmi publication, New Publication.
2. Programming in C by E. Balagurusamy TMH Publications.
3. C Programming - Kernighen Ritche
4. Programming with C – Y. Kanetkar.
5. C Programming – Holzner, PHI Publication.
6. Programming in C – Ravichandran.



C, computer programming language developed in the early 1970s by American computer scientist **Dennis M. Ritchie** at Bell Laboratories (formerly AT&T Bell Laboratories).

Introduction to Problem Solving Techniques

Structure

1. Introduction
2. Procedure (steps involved in problem solving)
3. Algorithm
4. Flow Chart
5. Symbols used in Flow Charts
6. Pseudo Code

Learning Objectives

- To understand the concept of Problem solving
- To understand steps involved in algorithm development
- To understand the concept of Algorithm
- Develop Algorithm for simple problem
- To understand the concept of Flowchart development
- Draw the symbols used in Flowcharts

1. 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. Apart 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.

1. 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. Understanding the problem
2. Analyzing the problem
3. Developing the solution
4. Coding and implementation.

1. Understanding the problem: Here we try to understand the problem to be solved in totally. Before with the next stage or step, we should be absolutely sure about the objectives of the given problem.

2. Analyzing the problem: After understanding thoroughly the problem to be solved, we look different ways of solving the problem and evaluate each

of these methods. The idea here is to search an appropriate solution to the problem under consideration. The end result of this stage is a broad overview of the sequence of operations that are to be carried out to solve the given problem.

3.Developing the solution: Here the overview of the sequence of operations that was the result of analysis stage is expanded to form a detailed step by step solution to the problem under consideration.

4.Coding and implementation: The last stage of the problem solving is the conversion of the detailed sequence of operations into a language that the computer can understand. Here each step is converted to its equivalent instruction or instructions in the computer language that has been chosen for the implementation.

1.2 Algorithm

Definition

A set of sequential steps usually written in Ordinary Language to solve a given problem is called **Algorithm**.

It may be possible to solve a problem in more than one way, resulting in more than one algorithm. The choice of various algorithms depends on the factors like reliability, accuracy and easy to modify. The most important factor in the choice of algorithm is the time requirement to execute it, after writing code in High-level language with the help of a computer. The algorithm which will need the least time when executed is considered the best.

Steps involved in algorithm development

An algorithm can be defined as “**a complete, unambiguous, finite number of logical steps for solving a specific problem**”

Step1. Identification of input: For an algorithm, there are quantities to be supplied called input and these are fed externally. The input is to be identified first for any specified problem.

Step2: Identification of output: From an algorithm, at least one quantity is produced, called for any specified problem.

Step3 : Identification the processing operations : All the calculations to be performed in order to lead to output from the input are to be identified in an orderly manner.

Step4 : Processing Definiteness : The instructions composing the algorithm must be clear and there should not be any ambiguity in them.

Step5 : Processing Finiteness : If we go through the algorithm, then for all cases, the algorithm should terminate after a finite number of steps.

Step6 : Possessing Effectiveness : The instructions in the algorithm must be sufficiently basic and in practice they can be carried out easily.

An algorithm must possess the following properties

1. **Finiteness**: An algorithm must terminate in a finite number of steps
2. **Definiteness**: Each step of the algorithm must be precisely and unambiguously stated
3. **Effectiveness**: Each step must be effective, in the sense that it should be primitive easily convertible into program statement) can be performed exactly in a finite amount of time.
4. **Generality**: The algorithm must be complete in itself so that it can be used to solve problems of a specific type for any input data.
5. **Input/output**: Each algorithm must take zero, one or more quantities as input data produce one or more output values. An algorithm can be written in English like sentences or in any standard representation sometimes, algorithm written in English like languages are called Pseudo Code

Example

1. Suppose we want to find the average of three numbers, the algorithm is as follows

Step 1 Read the numbers a, b, c

Step 2 Compute the sum of a, b and c

Step 3 Divide the sum by 3

Step 4 Store the result in variable d

Step 5 Print the value of d

Step 6 End of the program

2. Algorithms for Simple Problem Write an algorithm for the following

1. Write an algorithm to calculate the simple interest using the formula. Simple interest = $P \times N \times R / 100$.

Paper - II Programming in C

Where P is principle Amount, N is the number of years and R is the rate of interest.

Step 1: Read the three input quantities' P, N and R. Step 2 : Calculate simple interest as

Simple interest = $P * N * R / 100$

Step 3: Print simple interest. Step 4: Stop.

2.Area of Triangle: Write an algorithm to find the area of the triangle.

Let b, c be the sides of the triangle ABC and A the included angle between the given sides.

Step 1: Input the given elements of the triangle namely sides b, c and angle between the sides A.

Step 2: Area = $(1/2) * b * c * \sin A$ Step 3: Output the Area

Step 4: Stop.

3. Write an algorithm to find the largest of three numbers X, Y, Z. Step 1: Read the numbers X, Y, Z.

Step 2: if $(X > Y)$

Big = X

else BIG = Y

Step 3 : if $(BIG < Z)$ Step 4: Big = Z

Step 5: Print the largest number i.e. Big

Step 6: Stop.

4. Write down an algorithm to find the largest data value of a set of given data values

Algorithm largest of all data values:

Step 1: $LARGE \leftarrow 0$ Step 2: read NUM

Step 3: While $NUM \geq 0$ do

1. if $NUM > LARGE$

1. then

1. $LARGE \leftarrow NUM$

3.2. read NUM

Step 4: Write “largest data value is”, LARGE Step 5: end.

5. Write an algorithm which will test whether a given integer value is prime or not.

Algorithm prime testing:

Step 1: $M \leftarrow 2$ Step 2: read N

Step 3: $MAX \leftarrow \text{SQRT}(N)$

Step 4: While $M \leq MAX$ do

1. if $(M * (N/M) = N)$

1. then

4.1.1.1 go to step 7

4.2. $M \leftarrow M + 1$ Step 5: Write “number is prime” Step 6: go to step 8

Step 7: Write “number is not a prime” Step 8: end.

Paper - II Programming in C

6. Write algorithm to find the factorial of a given number N Step 1: $PROD \leftarrow 1$

Step 2: $I \leftarrow 0$ Step 3: read N

Step 4: While $I < N$ do

4.1 $I \leftarrow I + 1$

4.2. $PROD \leftarrow PROD * I$

Step 5: Write “Factorial of”, N, “is”, PROD Step 6: end.

7. Write an algorithm to find sum of given data values until negative value is entered.

Algorithm Find – Sum Step 1: $SUM \leftarrow 0$

Step 2: $I \leftarrow 0$

Step 3: read NEW VALUE

Step 4: While $NEW\ VALUE \leq 0$ do

4.1 $SUM \leftarrow SUM + NEW\ VALUE$ 4.2 $I \leftarrow I + 1$

4.3 read NEW VALUE

Step 5: Write “Sum of”, I, “data value is”, “SUM Step 6: END

8. Write an algorithm to calculate the perimeter and area of rectangle. Given its length and width.

Step 1: Read length of the rectangle. Step 2: Read width of the rectangle.

Step 3: Calculate perimeter of the rectangle using the formula $perimeter = 2 * (length + width)$

Step 4: Calculate area of the rectangle using the formula $area = length * width$.

Step 5: Print perimeter. Step 6: Print area.

Step 7: Stop.

3. Flowchart

A flow chart is a step by step diagrammatic representation of the logic paths to solve a given problem. Or Aflowchart is visualor graphical representation of an algorithm.

The flowcharts are pictorial representation of the methods to b used to solve a given problem and help a great deal to analyze the problem and plan its solution in a systematic and orderly manner. A flowchart when translated in to a proper computer language, results in a complete program.

Advantages of Flowcharts

1. The flowchart shows the logic of a problem displayed in pictorial fashion which facilitates easier checking of an algorithm.
2. The Flowchart is good means of communication to other users. It is also a compact means of recording an algorithm solution to a problem.
3. The flowchart allows the problem solver to break the problem into parts. These parts can be connected to make master chart.
4. The flowchart is a permanent record of the solution which can be consulted at a later time.

Differences between Algorithm and Flowchart

Algorithm	Flowchart
1. A method of representing the step-by-step logical procedure for solving a problem	1.Flowchart is diagrammatic representation of an algorithm. It is constructed using different types of boxes and symbols.
2. It contains step-by-step English descriptions, each step representing a particular operation leading to solution of problem	2.The flowchart employs a series of blocks and arrows, each of which represents a particular step in an algorithm
3. These are particularly useful for small problems	3.These are useful for detailed representations of complicated programs
4. For complex programs, algorithms prove to be Inadequate	4.For complex programs, Flowcharts prove to be adequate

4. 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).

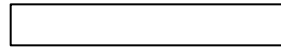
a. Oval: Rectangle with rounded sides is used to indicate either START/ STOP of the program. ..



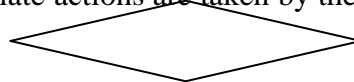
b. Input and output indicators: Parallelograms are used to represent input and output operations. Statements like INPUT, READ and PRINT are represented in these Parallelograms.



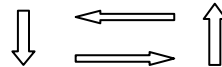
c. Process Indicators: - Rectangle is used to indicate any set of processing operation such as for storing arithmetic operations.



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. The decision box symbol is



e. Flow Lines: 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.

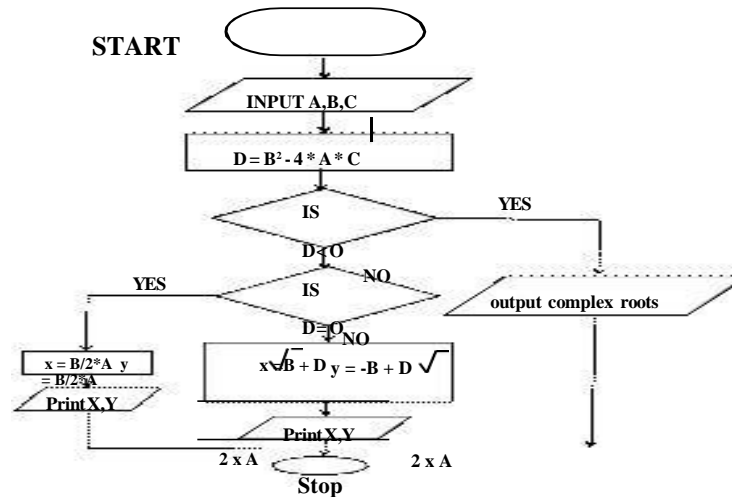


g. Off-page connectors: This connector represents a break in the path of flowchart which is too large to fit on a single page. It is similar to on-page connector. The connector symbol marks where the algorithm ends on the first page and where it continues on the second.



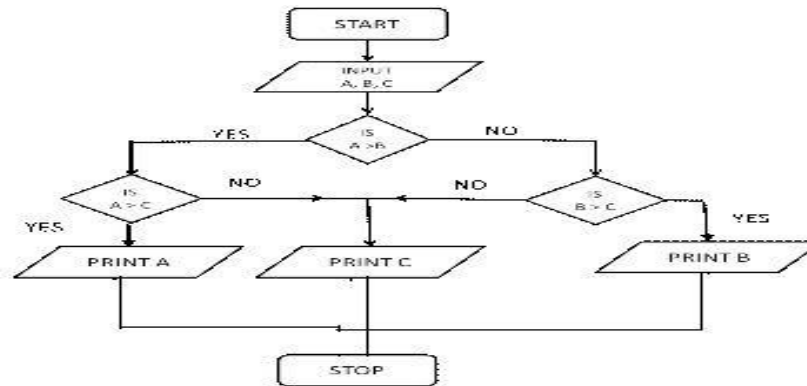
1.Simple Problems using Flow Chart Draw the Flowchart for the following

1. Draw the Flowchart to find Roots of Quadratic equation $ax^2 + bx + c = 0$. The coefficients a, b, c are the input data

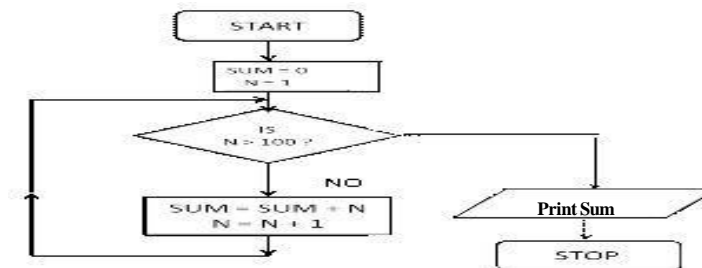


Paper - II Programming in C

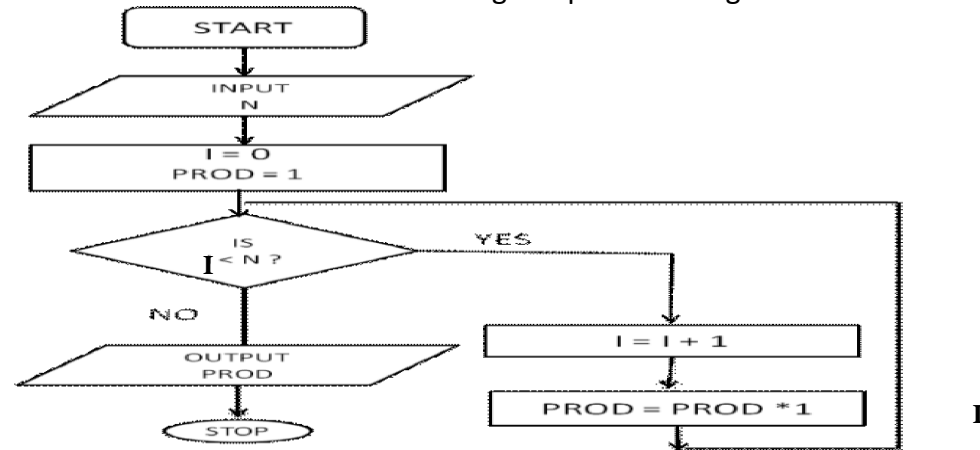
2. Draw a flowchart to find out the biggest of the three unequal positive numbers.



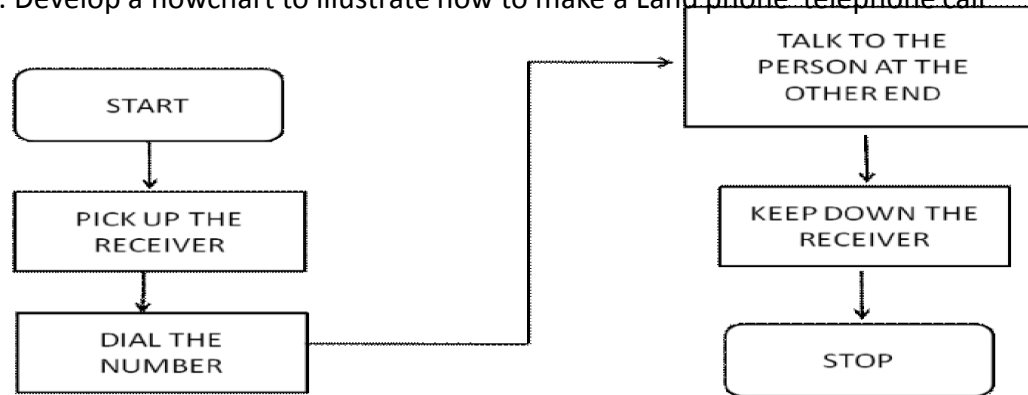
3. Draw a flowchart for adding the integers from 1 to 100 and to print the sum.



4. Draw a flowchart to find the factorial of given positive integer N.

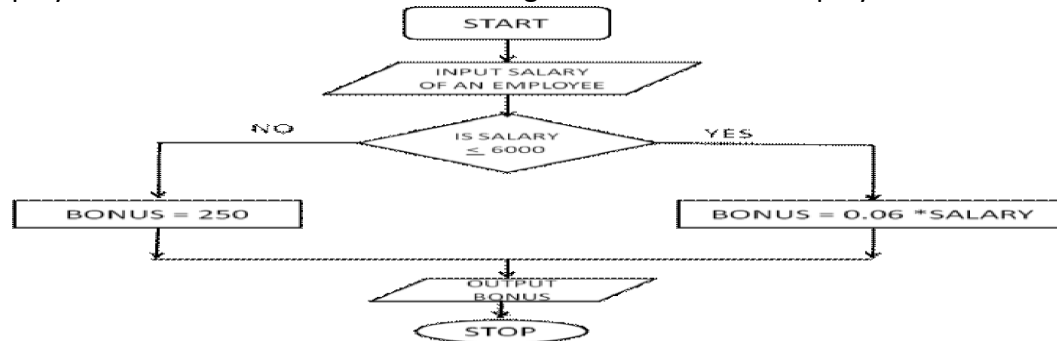


5. Develop a flowchart to illustrate how to make a Land phone telephone call



Flowchart for Telephone call

6. ABC company plans to give a 6% year-end bonus to each of its employees earning Rs 6,000 or more per month, and a fixed Rs 250/- bonus to the remaining employees. Draw a flowchart for calculating the bonus for an employee



1.5 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.

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.

Activity

Practice more sample problems on algorithm and Flowcharts

Model Questions

Short Answer Type Questions - 2 Marks

1. Define Algorithm
2. What is Flowchart
3. What is Pseudo code?
4. What are the symbols of Flowchart
5. Write an Algorithm for perimeter of Triangle
6. What are the basic steps involved In problem solving

Long Answer Type Questions - 6 Marks

1. Differentiate between Algorithm and Flowchart.
2. Write an algorithm to find greatest of given three numbers.
3. Write an algorithm to check whether given integer value is PRIME or NOT.
4. Draw the flowchart to find roots of Quadratic equation $ax^2 + bx + c = 0$ Note : Practice more related Algorithms and Flowcharts.

**Thank
you**