

Algorithms and Flowcharts

TOPIC – 1

Problem Solving Methodologies

Very Short Answer Type Questions (1 mark each)

Question 1:

Write the alternate name of infinite loop.

Answer:

endless loop.

Question 2:

Define looping

Answer:

Looping is repeating a s t of instructions until a specific condition is met

Question 3:

Write the difference between finite and infinite loop.

Answer:

A finite loop ends itself. An infinite loop will not end without an outside cause.

Question 4:

Is for loop pretest type of loop ?

Answer:

Yes. The second clause (the condition) is evaluated before each iteration through a loop.

Question 5:

Can you use one or more loop inside any another while, for or do., while loop ?

Answer:

Yes.

Question 6:

Write the name of loop which executes a sequence of statements multiple times and abbreviates the code that manages the loop variable.

Answer:

for-loop.

Question 7:

Define Variable.

Answer:

It is the name of space inside computer memory values can be stored.

Short Answer Type Questions-I (2 marks each)

Question 1:

Define infinite loop.

Answer:

An infinite loop is a sequence of instructions in a computer program which loops endlessly, either due to the loop having no terminating condition, having one that can never be met, or one that causes the loop to start over.

Question 2:

Give an example for infinite loop

Answer:

A simple example of an infinite loop is instructing a computer to keep on adding 0 to 1 until 2 is reached. This will never happen.

Question 3:

What is a logical operator ?

Answer:

Logical operator is a operator used to combine relational expressions. In C++, there are three types of logical operators :

- (i) Logical AND
- (ii) Logical OR
- (iii) Logical NOT

Question 4:

What will be the minimum number of inputs for finding a simple interest ?

Answer:

The formula of simple interest is :

$$SI = \frac{P \times R \times T}{100}$$

So, to find the simple interest we need to input the value of Principal (P), Rate (R) and Time (T). Therefore the minimum number of inputs for finding the simple interest is 3 i.e. P R and T.

Long Answer Type Questions (4 marks each)

Question 1:

Explain Modular programming.

Answer:

- (i) Modular programming is the technique which divides the entire program into smaller modules, which perform a specific task.
- (ii) It can often be used in variety of applications and functions with other components of the system.
- (iii) The order in which the modules are executed by the computer is controlled by the main program. This describes fully the procedures required in the solution to a problem. The procedures are written in the order of the machine execution.
- (iv) OPP is compatible with the modular programming concept to a large extent. Modular Programming enables multiple programmers to divide up the work and debing pieces of the program independently.

Question 2:

What is the difference between modular programming and structured programming ?

Answer:

Modular programming : It is the act of designing and writing programs as interactions among functions that each perform a single well defined function & which have minimal side effect interaction between them. It is heavily procedural. The focus is entirely on writing code (functions). Data is passive. Any code may access the contents of any data structured passed to it.

Object Oriented programming : it is a programming using “objects”-data structures consisting of data fields & methods together with their interactions to design applications and computer programs. Programming techniques may include features such as data abstraction, encapsulation, messaging, modularity, polymorphism/and inheritance.

Question 3:

Differentiate between top down and bottom up methods of modular programming.

Answer:

Top-Down Method	Bottom-up Method
The principle of top down method dictate that a program should be divided into a main module and its related module. Each module should also be divided into sub modules according to software engineering and programming style. The division continues till the module consists only of an elementary process that is intrinsically understood and cannot be further subdivided.	Bottom-up method is just the opposite of top- down method. It refers to a style of programming in which an application is constructed with existing primitives of the programming language and then gradually more and more complicated features are added till applications are written. In other words, initiating the design with simple modules and then build them into more complex structures ending at the top is bottom-up method.

Question 4:

What are the advantages of using a modular approach in programming ?

Answer:

- It is easy to understand small sections of code in the modular programming.
- You can store the code across multiple files.
- Modular programming allows collaborative programming. Collaborative programming means more' than one program work in one application at the same time.
- Duplication of code, is not possible in modular programming. The programmers create a single procedure for code.
- The errors are localized to a subroutine or function and it is easy to find the errors.
- The code should be used in multiple applications in modular programming.
- The code should be simple and short in modular programming and it is less need to be written.

TOPIC – 2

Algorithms & Flowcharts

Short Answer Type Questions-II (3 marks each)

Question 1:

Explain Algorithm.

Answer:

An algorithm is a step by step instructions required to solve any problem. These instructions are in user's language that means they are independent of any programming language. **3**

Question 2:

Write some properties of algorithm.

Answer:

Some properties of algorithm are :

1. An algorithm should terminate after finite number of steps.
2. An algorithm should be simple so that anyone can understand them properly.
3. An algorithm should be universal and lead to an unique solution of the problem.
4. An algorithm should have capability to handle some unexpected situations which may arise during the solution of a particular problem.
5. An algorithm should be able to solve all problems of a particular type for which it is designed. **[Any 3,1 mark for each correct point]**

Question 3:

Write some characteristics of algorithm.

Answer:

Some characteristics of algorithm are :

1. Each and every instruction should be short and clear.
2. Each instruction should be such that it can be performed in a finite time.
3. One or more instructions should not be repeated infinitely.
4. After performing the instructions the desired result must be obtained. **[Any 3,1 mark for each correct point]**

Question 4:

Write an algorithm to calculate sum of two numbers.

Answer:

Step 1 : Start.

Step 2 : Read A,B.

Step 3: Sum=A+B.

Step 4 : Print Sum.

Step 5 : Stop. **3**

Question 5:

Write an algorithm to calculate simple interest.

Answer:

Step 1: Start.

Step 2 : Read RR,T.

Step 3 : $SI = (P * R * T) / 100$.

Step 4 : Print SI

Step 5 : Stop. **3**

Question 6:

Write an algorithm to convert temperature from Fahrenheit to Celsius.

Answer:

Step 1: Start.

Step 2 : Read F.

Step 3: $C = (5(F - 32)) / 9$.

Step 4 : Print C.

Step 5 : Stop. **3**

Question 7:

Write an algorithm to calculate area of triangle.

Answer:

Step 1: Start.

Step 2 : Read a,b,c.

Step 3: $s = (a + b + c) / 2$.

Step 4 : Area = $(s(s-a)(s-b)(s-c))^{1/2}$.

Step 5 : Print Area.

Step 6 : Stop. **3**

Question 8:

Write an algorithm that generates the fibonacci series as:

1,1,2,3, 5,.... N terms.

Answer:

The algorithm as :

Step 1 : Begin.
Step 2 : Read N.
Step 3 : Let $a = 1$, $b = 1$ Step 4 : Print a, b.
Step 5 : $ctr = 2$.
Step 6 : $c = a + b$.
Step 7 : $a = b$.
Step 8 : $b = c$.
Step 9 : $ctr = ctr + 1$.
Step 10 : If $ctr < N$ then repeat steps 6 to 9.
Step 11 : Stop. **3**

Question 9:

Write an algorithm to compute sum of the square of N numbers.

Answer:

The algorithm as :

Step 1: Read N.
Step 2 : Let $ctr = 0$, $sum = 0$.
Step 3 : Read Num.
Step 4 : $ctr = ctr + 1$.
Step 5 : Compute the square of the number *i.e.*,
 $= \text{sqr}(\text{Num} * \text{Num})$.
Step 6 : $sum = sum + \text{sqr}$.
Step 7 : If ctr is less than N then repeat steps 3 to 6. Step 8 : Print sum.
Step 9 : End. **3**

Question 10:

Write an algorithm for calculating the conversion from rupees to dollars.

Answer:

Algorithm is defined as follows :

Step 1: Start.
Step 2: Read the amount in rupees and conversion rate.
Step 3: Calculate the equivalent amount in dollars using the following formula : amount in dollars = amount in rupees / conversion rate.
Step 4 : Write the amount in dollars.
Step 5 : Stop. **3**

Question 11:

Write an algorithm to find the largest among three numbers.

Answer:

The algorithm is defined as :

Step 1: Read X, Y, Z.

Step 2 : If $X > Y$ continue step 5.

Step 3 : If $Y > Z$ then print "Y is the largest "and continue step 7.

Step 4 : Continue step 6.

Step 5 : If $X > Z$ then print "X is the largest "and continue step 7.

Step 6 : Print "Z is largest".

Step 7: End. **3**

Question 12:

Write an algorithm to print the bigger of any two unique given numbers.

Answer:

The algorithm as :

Step 1: Read two numbers A and B.

Step 2 : Compare A and B.

Step 3 : If A is greater than B then, print A else print B.

Step 4 : Stop. **3**

Question 13:

Write an algorithm to input N numbers and find the largest among them.

Answer:

The algorithm as :

Step 1 : Read N.

Step 2 : Let $ctr = 1$.

Step 3 : Read Num.

Step 4 : $large = Num$.

Step 5 : Read next Num.

Step 6 : $ctr = ctr + 1$.

Step 7 : If $(Num > large)$ then $large = Num$.

Step 8 : If $(ctr \leq N)$ then repeat steps 5 and 6.

Step 9 : Print large.

Step 10 : End. **3**

Question 14:

Write an algorithm to compute the sum of odd and even numbers up to N.

Answer:

The algorithm as :

Step 1: Start.

Step 2 : Input N.

Step 3 : $sum_e = 0$.

Step 4 : $sum_o = 0$.

Step 5 : $ctr = 1$.

Step 6 : while ($ctr \leq N$).

If ($ctr \bmod 2 = 0$)

$sum_e = sum_e + ctr$

else

$sum_o = sum_o + ctr$.

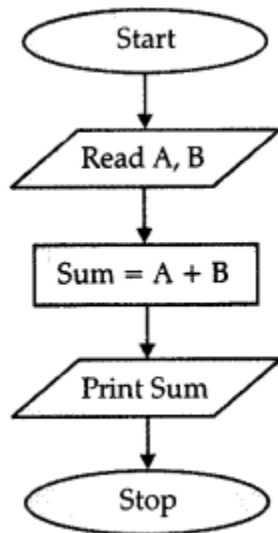
Step 7: Print "sum of even number is" sum_e .

Step 8 : Print "sum of odd number is" sum_o .

Step 9: End.

Question 1:

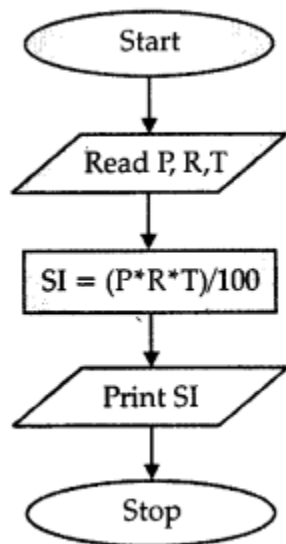
Draw a flowchart to calculate sum of two numbers

Answer:

Question 2:

Draw a flow chart to calculate simple interest

Answer:

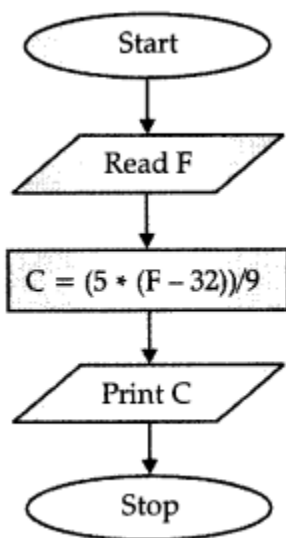


3

Question 3:

Draw flowchart to convert temperature from Fahrenheit to Celsius.

Answer:

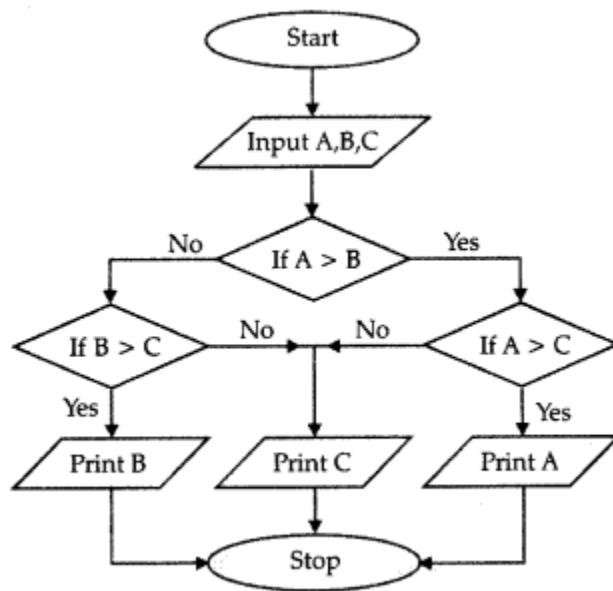


3

Question 4:

Draw the flowchart to print the largest of any three numbers.

Answer:

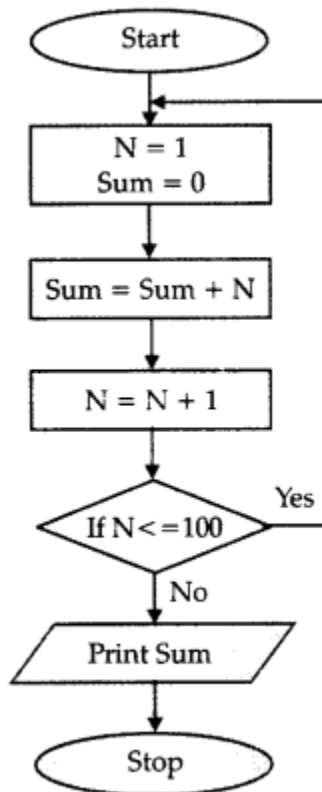


3

Question 5:

Draw the flowchart to print sum of first 100 natural numbers.

Answer:



3

