

Q1. Define Problem Solving and List out the steps of problem solving.

Ans: Problem Solving is the systematic process of understanding a problem, identifying possible solutions, and selecting and implementing the best solution to resolve the problem effectively.

Steps of Problem Solving:

1. Identify the Problem – Clearly understand and define what the problem is.
2. Analyze the Problem – Gather relevant information and understand the causes and constraints of the problem.
3. Generate Possible Solutions – Think of different ways to solve the problem.
4. Evaluate Alternatives – Compare the possible solutions and choose the most suitable one.
5. Implement the Solution – Put the chosen solution into action.
6. Review and Evaluate – Check whether the solution has solved the problem and make improvements if needed.

Q2. Define Algorithm. Explain the characteristics of an algorithm.

Ans: Definition of Algorithm: An algorithm is a finite, step-by-step set of clear and unambiguous instructions used to solve a problem or perform a specific task. Algorithms are commonly used in computer science to design programs and solve computational problems.

Characteristics of an Algorithm:

1. Input:
An algorithm should have zero or more inputs that are clearly defined.
2. Output:
An algorithm must produce at least one output as a result of processing the input.
3. Definiteness:
Each step of the algorithm should be precise, clear, and unambiguous.

4. Finiteness:

The algorithm must terminate after a finite number of steps.

5. Effectiveness:

Each step of the algorithm should be simple and executable in a finite amount of time.

6. Generality:

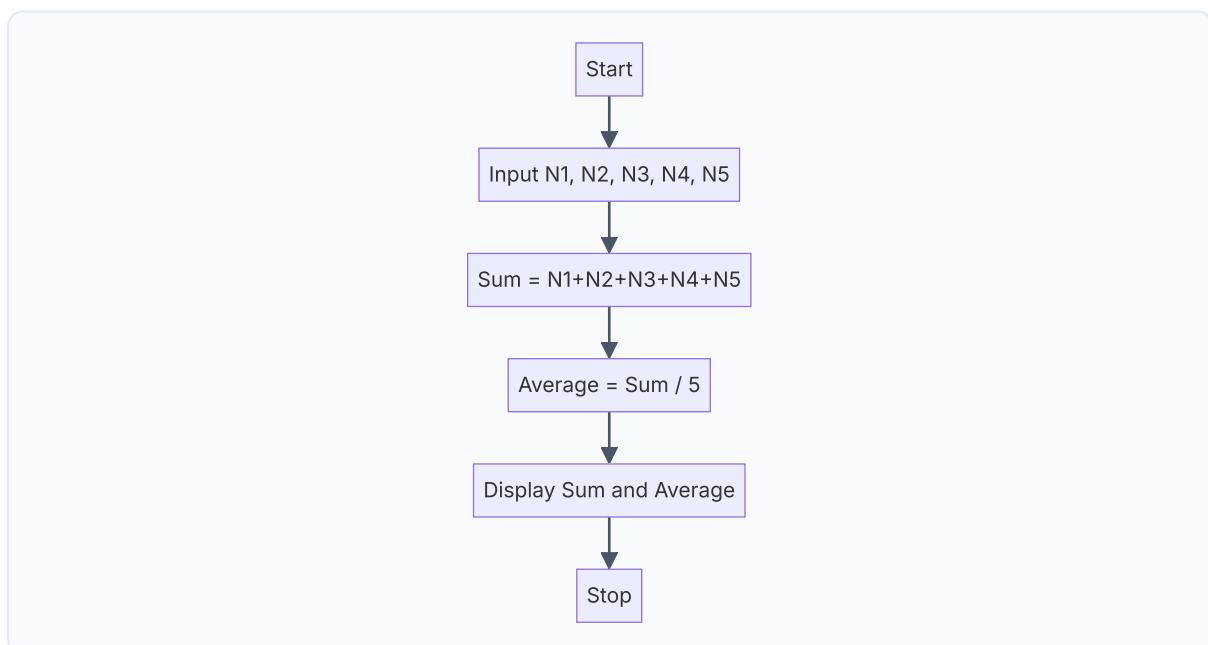
An algorithm should be applicable to all valid inputs of a particular problem, not just a specific case.

Q3. Write an algorithm and draw flowchart to find sum and average of any 5 numbers.

Answer: Algorithm to Find Sum and Average of Any 5 Numbers:

- Step 1: Start
- Step 2: Read five numbers: N1, N2, N3, N4, N5
- Step 3: Calculate Sum = N1 + N2 + N3 + N4 + N5
- Step 4: Calculate Average = Sum / 5
- Step 5: Display Sum and Average
- Step 6: Stop

Flowchart to Find Sum and Average of 5 Numbers:



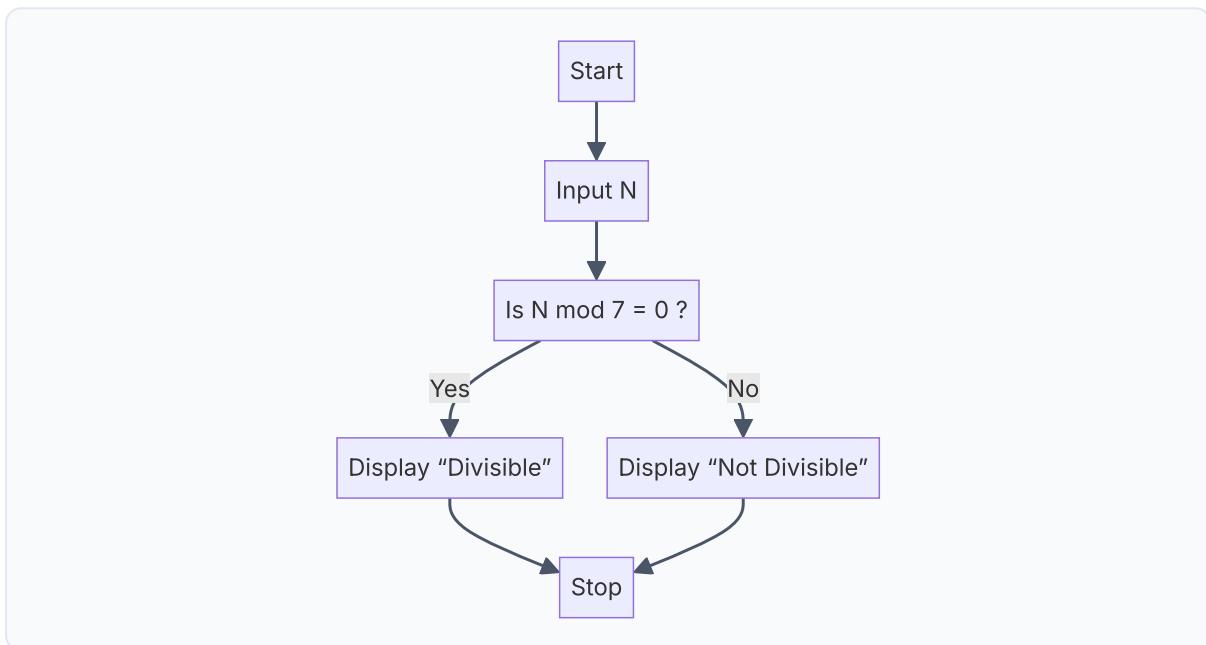
This algorithm and flowchart clearly show how to calculate the sum and average of any five numbers step by step.

Q4. Write an algorithm and draw flowchart to check entered number is divisible by 7 or not.

Ansver: Algorithm to Check Whether a Number is Divisible by 7:

- Step 1: Start
- Step 2: Read a number N
- Step 3: If $N \bmod 7 = 0$, then go to Step 4, else go to Step 5
- Step 4: Display "Number is divisible by 7"
- Step 5: Display "Number is not divisible by 7"
- Step 6: Stop

Flowchart to Check Divisibility by 7:



This algorithm and flowchart help in clearly determining whether an entered number is divisible by 7 or not.

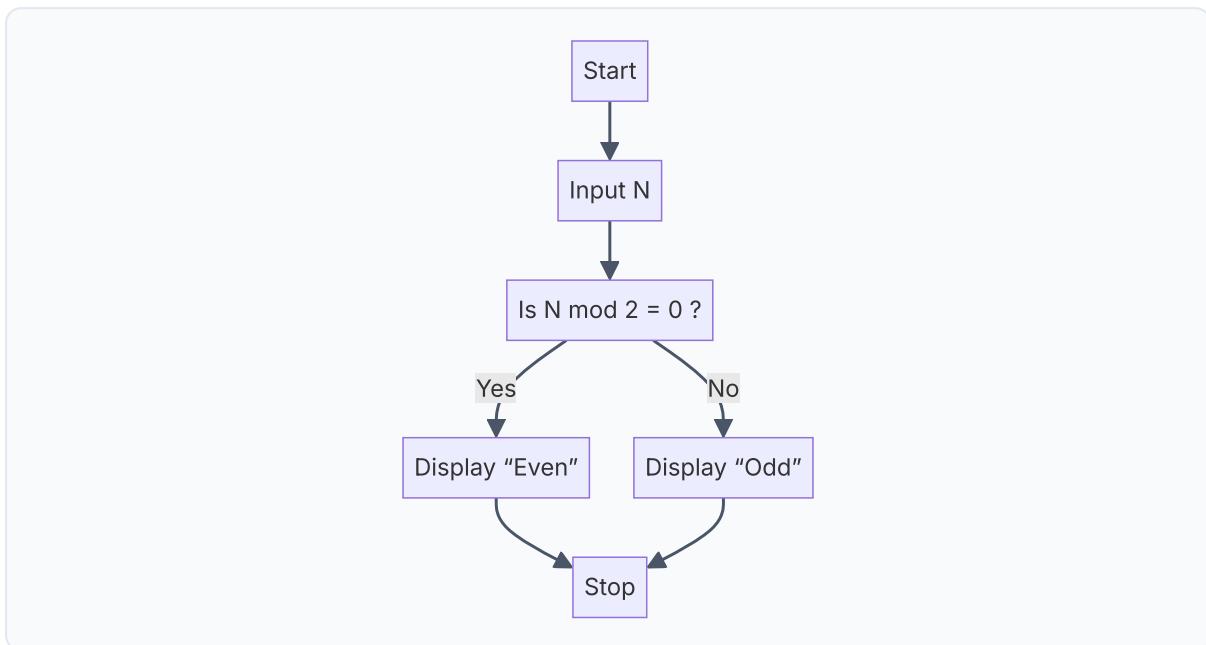
Q5. Write an algorithm and draw flowchart to check entered number is even or odd.

Ansver: Algorithm to Check Whether a Number is Even or Odd:

- Step 1: Start
- Step 2: Read a number N
- Step 3: If $N \bmod 2 = 0$, then go to Step 4, else go to Step 5
- Step 4: Display "Number is Even"
- Step 5: Display "Number is Odd"

Step 6: Stop

Flowchart to Check Even or Odd:



This algorithm and flowchart clearly show how to determine whether the entered number is even or odd.

Q6. Write an algorithm and draw flowchart to find minimum of two numbers.

Answer: Algorithm to Find the Minimum of Two Numbers:

Step 1: Start

Step 2: Read two numbers A and B

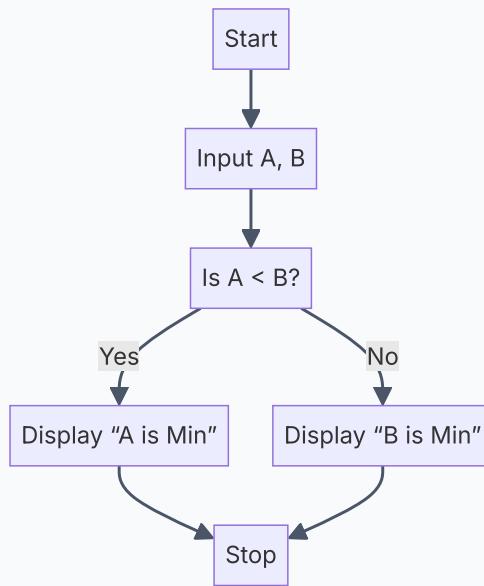
Step 3: If $A < B$, then go to Step 4, else go to Step 5

Step 4: Display "A is the minimum number"

Step 5: Display "B is the minimum number"

Step 6: Stop

Flowchart to Find Minimum of Two Numbers:



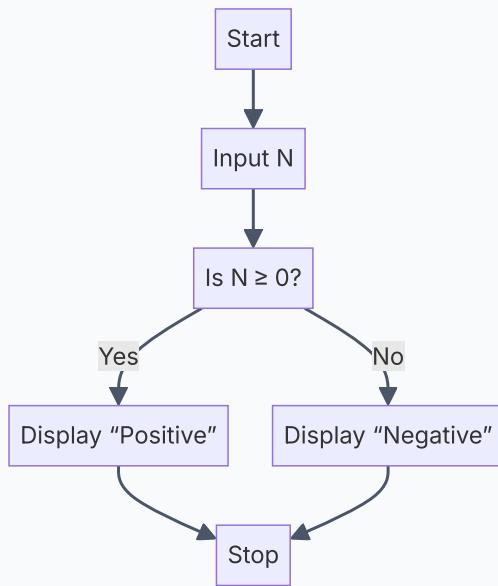
This algorithm and flowchart clearly explain how to find the minimum of two numbers.

Q7. Write algorithm and draw flowchart to find entered number is positive or negative.

Ans: Algorithm to Check Whether a Number is Positive or Negative:

- Step 1: Start
- Step 2: Read a number N
- Step 3: If $N \geq 0$, then go to Step 4, else go to Step 5
- Step 4: Display "Number is Positive"
- Step 5: Display "Number is Negative"
- Step 6: Stop

Flowchart to Check Positive or Negative Number:



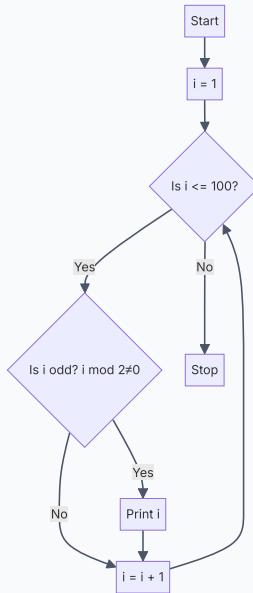
This algorithm and flowchart help in identifying whether the entered number is positive or negative.

Q8. Write algorithm and draw flowchart to print odd numbers between 1 to 100.

Ansver: Algorithm to Print Odd Numbers from 1 to 100:

- Step 1: Start
- Step 2: Set $i = 1$
- Step 3: If $i \leq 100$, go to Step 4, else go to Step 6
- Step 4: If $i \bmod 2 \neq 0$, display i
- Step 5: Increment $i = i + 1$ and go back to Step 3
- Step 6: Stop

Flowchart to Print Odd Numbers from 1 to 100:



This algorithm and flowchart clearly show how to print all odd numbers between 1 and 100.

Q9. Write a program and draw flowchart to calculate simple interest using below equation. $I=PRN/100$, Where P=Principal amount, R=Rate of interest and N=Period

Ans: Program and Flowchart to Calculate Simple Interest.

Formula:

- $I = \frac{P \times R \times N}{100}$

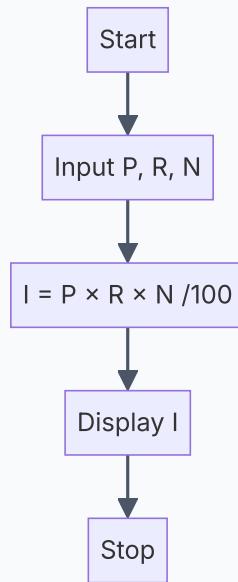
Where:

- P = Principal amount
- R = Rate of interest
- N = Period (Time)

Algorithm:

- Step 1: Start
- Step 2: Read values of P, R, and N
- Step 3: Calculate $I = (P \times R \times N) / 100$
- Step 4: Display the value of I
- Step 5: Stop

Flowchart to Calculate Simple Interest:



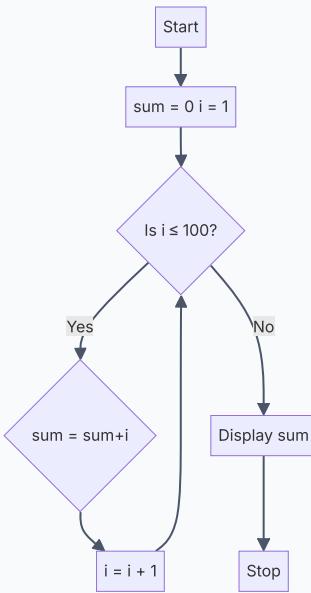
This program and flowchart clearly explain how to calculate Simple Interest using the given formula.

Q10. Write algorithm and draw flowchart to print sum of 1 to 100 numbers

Anser: Algorithm to Find the Sum of Numbers from 1 to 100:

- Step 1: Start
- Step 2: Set sum = 0 and i = 1
- Step 3: If $i \leq 100$, go to Step 4, else go to Step 6
- Step 4: Add i to sum → sum = sum + i
- Step 5: Increment $i = i + 1$ and go back to Step 3
- Step 6: Display the value of sum
- Step 7: Stop

Flowchart to Find Sum of Numbers from 1 to 100:



This algorithm and flowchart clearly show how to calculate and print the sum of numbers from 1 to 100.

Q11. Describe various types of control structures for algorithm and flowchart.

Answer: In programming and algorithm design, control structures define the order in which instructions are executed. They are essential for controlling the flow of a program or algorithm. Here's a detailed explanation:

Types of Control Structures:

1. Sequential Control Structure

- Definition: Instructions are executed one after another in sequence.
- Characteristics:
 - The simplest control structure.
 - No decision-making or repetition.
- Example:

Step 1: Read A
Step 2: Read B
Step 3: Sum = A + B
Step 4: Display Sum

- Flowchart Symbol: Straight sequence of processes connected by arrows.

2. Decision / Selection Control Structure

- Definition: Execution depends on a condition; it allows the program to choose between two or more paths.
- Types:
 - If...Then – Executes a block if the condition is true.
 - If...Then...Else – Executes one block if true, another if false.
 - Switch / Case – Selects from multiple options based on a value.
- Example:

```
If A > B Then
    Display "A is greater"
Else
    Display "B is greater"
End If
```

- Flowchart Symbol: Diamond shape for condition, arrows for Yes/No paths.

3. Loop / Iteration Control Structure

- Definition: Repeats a block of instructions until a condition is met.
- Types:
 - For Loop: Repeats a fixed number of times.
 - While Loop: Repeats while a condition is true.
 - Do...While Loop: Repeats at least once and continues while a condition is true.
- Example:

```
i = 1
While i ≤ 5
    Display i
    i = i + 1
End While
```

- Flowchart Symbol: Diamond for condition, arrows looping back to previous step.

4. Case / Multi-way Selection Control Structure

- Definition: Selects one path among multiple alternatives based on a value or expression.
- Example: Determine day of the week

```
Switch(day)
```

```

Case 1: Display "Sunday"
Case 2: Display "Monday"
...
End Switch

```

- Flowchart Symbol: Diamond for decision, multiple arrows for each case.

Summary Table

Control Structure	Purpose	Flow Representation
Sequential	Execute steps one after another	Straight arrows
Decision/Selection	Make choices based on condition	Diamond (Yes/No)
Loop/Iteration	Repeat steps until condition	Diamond with loop back
Case/Multi-way	Select from multiple options	Diamond with multiple branches

Q12. Define Pseudo Code and write the advantages of pseudo code.

Answer: Definition of Pseudo Code:

Pseudo code is a high-level description of an algorithm that uses plain English statements (or a mixture of English and programming-like keywords) to represent the steps of a program. It is not a real programming language, but it helps programmers design and plan the logic of a program before actual coding.

- Example of Pseudo Code:

```

START
    Read number N
    If N is even Then
        Display "Number is Even"
    Else
        Display "Number is Odd"
    End If
STOP

```

Advantages of Pseudo Code:

- Easy to Understand

- It uses plain English, so even non-programmers can understand the logic.

2. Language Independent

- Pseudo code is not tied to any programming language, so it can be used as a blueprint for any language.

3. Reduces Errors

- Helps programmers identify logical errors before writing actual code.

4. Simplifies Complex Problems

- Breaks down a problem into simple, manageable steps.

5. Improves Communication

- Useful for communicating ideas among team members or documenting algorithms.

6. Easy to Modify

- Changes in logic can be applied easily without worrying about syntax.

Q13. Write Pseudocode_(1) To compute the area of a circle (2) to find sum and average of 3 numbers (3) to find maximum of three given numbers (4) to solve the equation $f=4x^2+3y-8$

Ans: Here is a clear set of pseudo code examples for all four functions mentioned:

1 Pseudo Code to Compute the Area of a Circle:

```

START
    Read radius R
    Area = 3.14159 * R * R
    Display "Area of Circle is", Area
STOP

```

2 Pseudo Code to Find Sum and Average of 3 Numbers:

```

START
    Read A, B, C
    Sum = A + B + C
    Average = Sum / 3
    Display "Sum =", Sum
    Display "Average =", Average

```

STOP

3 Pseudo Code to Find Maximum of Three Given Numbers:

```
START
    Read A, B, C
    If A > B AND A > C Then
        Max = A
    Else If B > C Then
        Max = B
    Else
        Max = C
    End If
    Display "Maximum Number =", Max
STOP
```

4 Pseudo Code to Solve the Equation $f = 4x + 3y - 8$:

```
START
    Read x, y
    f = 4 * x + 3 * y - 8
    Display "Value of f =", f
STOP
```

These pseudo codes are language-independent and can easily be converted into actual programs or flowcharts.