



# Problem Solving: Flowchart and Algorithm

## Introduction

From the stone age to modern technology era, human come across the wide variety of problems every day throughout the life. It includes very simple tasks like counting different items, making total of bill for purchase of different items in different quantity having different prices etc. to complex problems like managing the stock of the hundreds of items in a store of a large factory producing home-use items. People are used to with how to solve their day to day and simple problems but when it comes to large and complex problems where completion in time with accuracy is mandatory, computer comes to the help.

Computer science provides us the convenient way to solve the problems or we can say the problem solving is at the heart of the computer science. It can be done by writing a program for the task we want to handle. Once a program is written for the task, it can be used for long time without solving problem again and again. For example, performing the banking operations, selecting the item online or ordering the items online and making the payment online using UPI (Unified Payment Interface).

In this chapter, we will be focusing on understanding the concepts of problem solving and then learning two widely used methods 1) Flowchart and 2) Algorithm with number of examples. After learning the chapter students will be able to understand the given problem statement and develop the flowchart and/or writing algorithm to solve the problem with clarity.

## Overview of Problem Solving

Problem solving is a step-wise solution of a problem. That means, understanding the problem completely and organizing it into smaller steps in proper order. The broader steps for problem solving can be as under:

- Define the problem with all details
- Understanding the problem
- Identifying or getting the inputs
- Providing the necessary steps to solve the problem
- Testing and validating the output

The step “Providing the necessary steps to solve the problem” varies from problem to problem. For simple problem, it includes few steps in sequence only but for moderate to large problems, it may include decisions providing alternative paths or repeating some steps again and again. Hence, this step mainly incorporates the following steps with different combinations:

- **Sequence** : one by one step
- **Decision making** : Decide to follow one of the steps from two more alternatives
- **Repetition/Looping** : Performing set of steps again and again i.e. multiple times

Let us understand them with the help of examples. The steps for adding two numbers are as follows:

1. Accept two numbers N1 and N2
2. Add them as  $SUM = N1 + N2$
3. Print the result i.e. SUM

The above problem is purely sequential and to solve it, we have to follow the steps in sequence as 1, 2 and 3.

Following are the steps for finding larger of two numbers.

1. Accept two numbers N1 and N2
2. Compare them and if  $N1 > N2$ , then N1 is larger, jump to step-4
3. Otherwise N2 is larger
4. Print larger

Observe that above problem includes decision at step-2. After accepting two numbers N1 and N2 in step-1, step-2 first compares them with “>” and if N1 is found larger, then it skips the step-3 and goes directly to step-4 to print N1. If the comparison is false, rest of the part of step-2 is skipped as N2 is larger means step-3 is performed and then step-4 prints N2. The important thing is we choose between two alternatives based on comparison of N1 or N2.

Following are the steps for making the sum of 1 to 10.

1. Define SUM with 0
2. Define I with 1
3. Repeat step-4 and step-5, 10 times
4. Add I to the SUM
5. Increment I by 1
6. Print SUM

Carefully, observe the steps of above problem. After step-1 defines SUM with 0 and step-2 defines I with first value 1, we repeat step-4 to add current value of I to the SUM and then increment I to next value in step-5. They are repeated 10 time hence I will go from 1,2, ... 10 and every time it is added to previous value of SUM. Finally, it produces  $SUM = 1+2+3+4+5+6+7+8+9+10 = 55$  which is printed by step-6.

Above example demonstrated use of all three important elements of processing steps of the problem solving. The problems we encounter in real-life are not simple as above. They are involving many computations and are complex. To solve them, we have to use above steps i.e. sequence, decision making and repetition in different combinations depending on the requirement of the problem.

Following section will introduce you with two widely used problem solving methods namely flowchart and algorithm. We will understand how to use them to solve the problem with many examples involving use of above steps and finally, we will compare both flowcharts and algorithms.

## Flowchart and its symbols

Let us understand the flowchart in detail with the different symbols used to develop the flowcharts with their meaning and application.

## Flowchart

Flowchart is a pictorial representation of the solution of a problem. It shows the steps of a process or task in a simple and visual way. Flowchart uses different symbols like ovals, rectangles, diamonds to represent various actions for the various steps. Arrows are used to connect different steps. As flowchart uses visual representation, all the possible paths and steps are clearly visible and it makes the understanding of the solution very easy. Figure 5.1 shows the flowchart for multiplying two numbers.

As shown in figure 5.1, flowchart starts with “Start” and ends with “End”. Second step represented by parallelogram accepts the two numbers N1 and N2. Third step uses rectangle to show the process of multiplying them as  $Mul = N1 * N2$ . Forth step is again parallelogram for printing the output i.e. Mul that is multiplication of given two numbers. The flowchart is very simple as problem is sequential. The flow i.e. order of the steps is clearly visible as arrow after each step points to the next step. Let us understand all the symbols used by flowchart before we develop the flowchart for different problems involving decision making and looping.

### Flowchart Symbols

Figure 5.2 shows the various symbols used to develop the flowcharts. Figure shows the symbol, its name and the short description for quick understanding. Let us now understand each of them one by one with their use in flowchart.

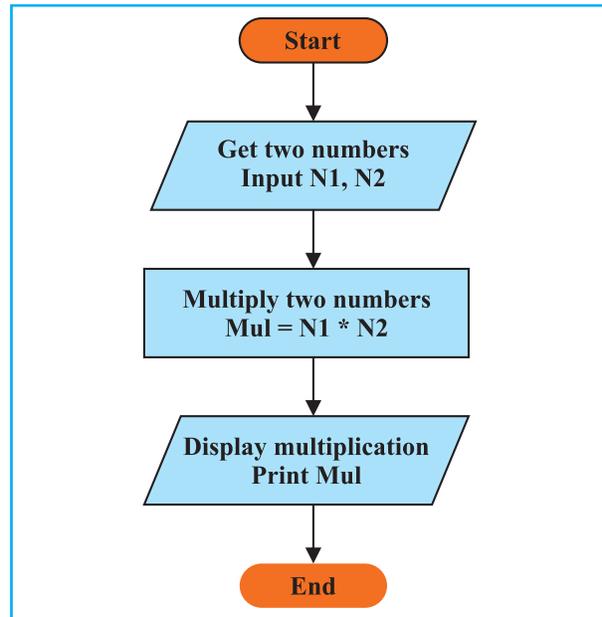


Figure 5.1 : Flowchart to Multiply Two Numbers

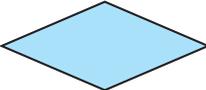
Symbol	Name	Description
	Oval	Represents start or end point
	Arrows	Connect two symbols with direction
	Parallelogram	Represents input or output
	Rectangle	Represents process, used to compute values or perform operations
	Diamond	Represents decision making, provides alternative paths
	Circle	Used as connector, used to connect different parts of flowchart

Figure 5.2 : Flowchart Symbols

**Start/End:** Start and End are the terminal symbols represented by oval. The Start denotes the beginning of the flowchart and is always the first symbol. The End denotes the completion or end of the flowchart and is always the last symbol. The steps for performing process or task are enclosed between Start and End symbols as shown in figure 5.1. Figure 5.3 shows the use of the Start and End symbol with arrows.

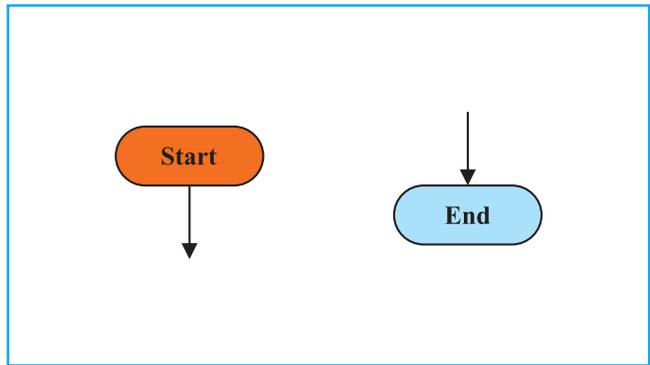


Figure 5.3 : Use of Start and End symbols

**Arrow:** Arrow is used to denote the flow lines. We can use arrow in all the directions depending on the where next symbol is drawn in the flowchart. Most common use is down arrow as shown in figure 5.1. Arrow is most important symbol used to decide the execution path within the flowchart or it indicates where the control moves next.

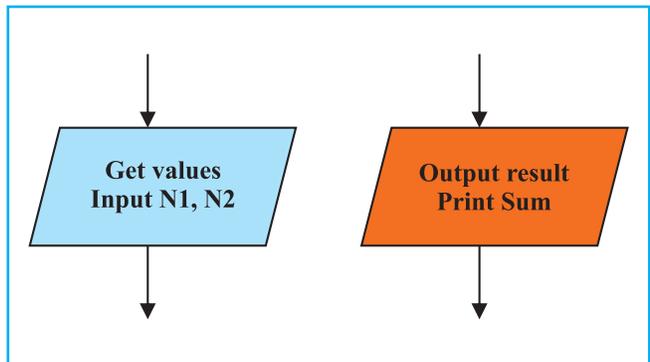


Figure 5.4 : Use of Input/Output symbols

However, arrow to left or right is also used whenever needed specifically in large flowcharts having many steps in a single page.

**Input/Output:** Input and Output are represented by parallelogram as shown in figure 5.2. The Input symbol is used to receive the values or data from the user or real world on which further process is done. The Output symbol denotes that the result produced by the solution is sent to the user or real world for their use. Figure 5.4 shows the use of the Input and Output symbols. Input symbol is denoted with “Input N1 and N2” and Output symbol denoted by “Print Sum”.

**Process:** It uses the rectangle. Whenever we want to compute or process some values, it is used. Process step can be used to perform variety of computations like arithmetic operations, logical operations, evaluating formulas based on given values etc. It is possible to compute

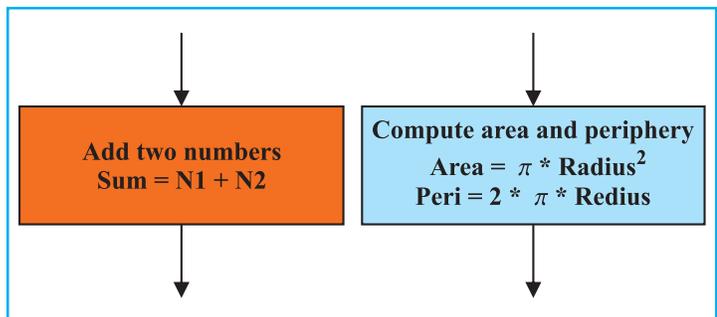


Figure 5.5 : Use of Process Symbol

more than one value using set of steps in a single process box represented by a rectangle. Figure 5.5 shows two examples of process step: one to add two numbers ( $Sum = N1 + N2$ ) and second to compute area and periphery of the circle.

**Decision:** Decision making is most important and performed regularly by human beings in their life. Decision uses the diamond symbol in the flowchart. Within a diamond, we normally use condition to decide one of the alternatives. Let us say, if number is greater than 0, we negate it, otherwise don't take any action. Another example compares two numbers and if first is greater than second, we take choice-1, otherwise we take choice-2. It is shown in figure 5.6.

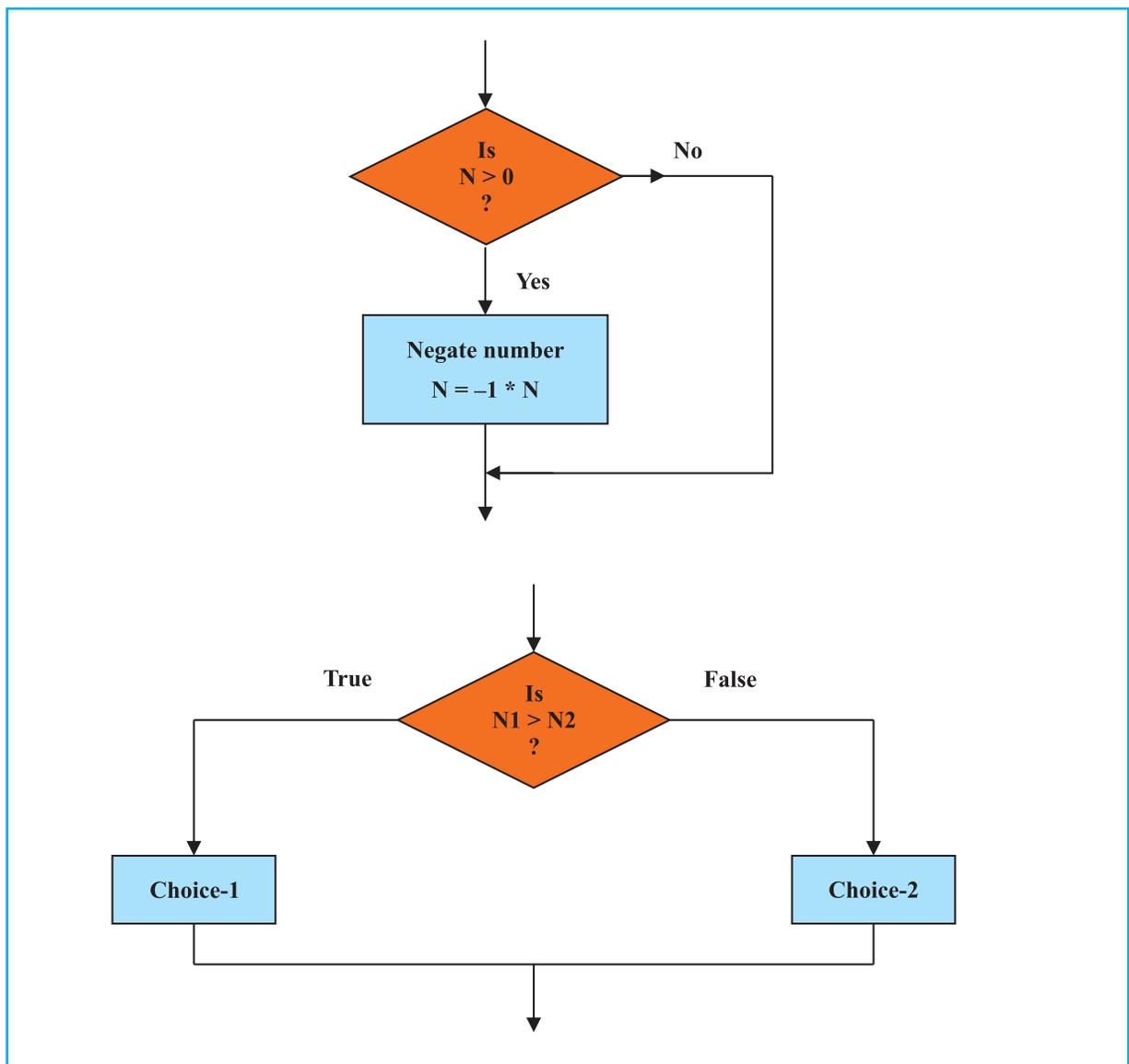


Figure 5.6 : Use of Decision Making

Sometimes, we have multiple conditions and with each of them, some statements are associated. Figure 5.7 shows such a situation. It is a multi-choice structure as we have to choose one alternative from multiple alternatives based on which one of the conditions is true. In figure 5.7, first the Condition1 is tested. If it is true, then statement1 is processed and control moves to the end of structure. If it is false, then Condition2 is evaluated. If Condition2 is true, then statement2 is processed and moves to the end, otherwise it moves to the Condition3. It is repeated until all the conditions are tested. If all the conditions are false, then Default-statement is processed and completes. In short, only one of the statements will execute depending on which condition is true or Default-statement executes.

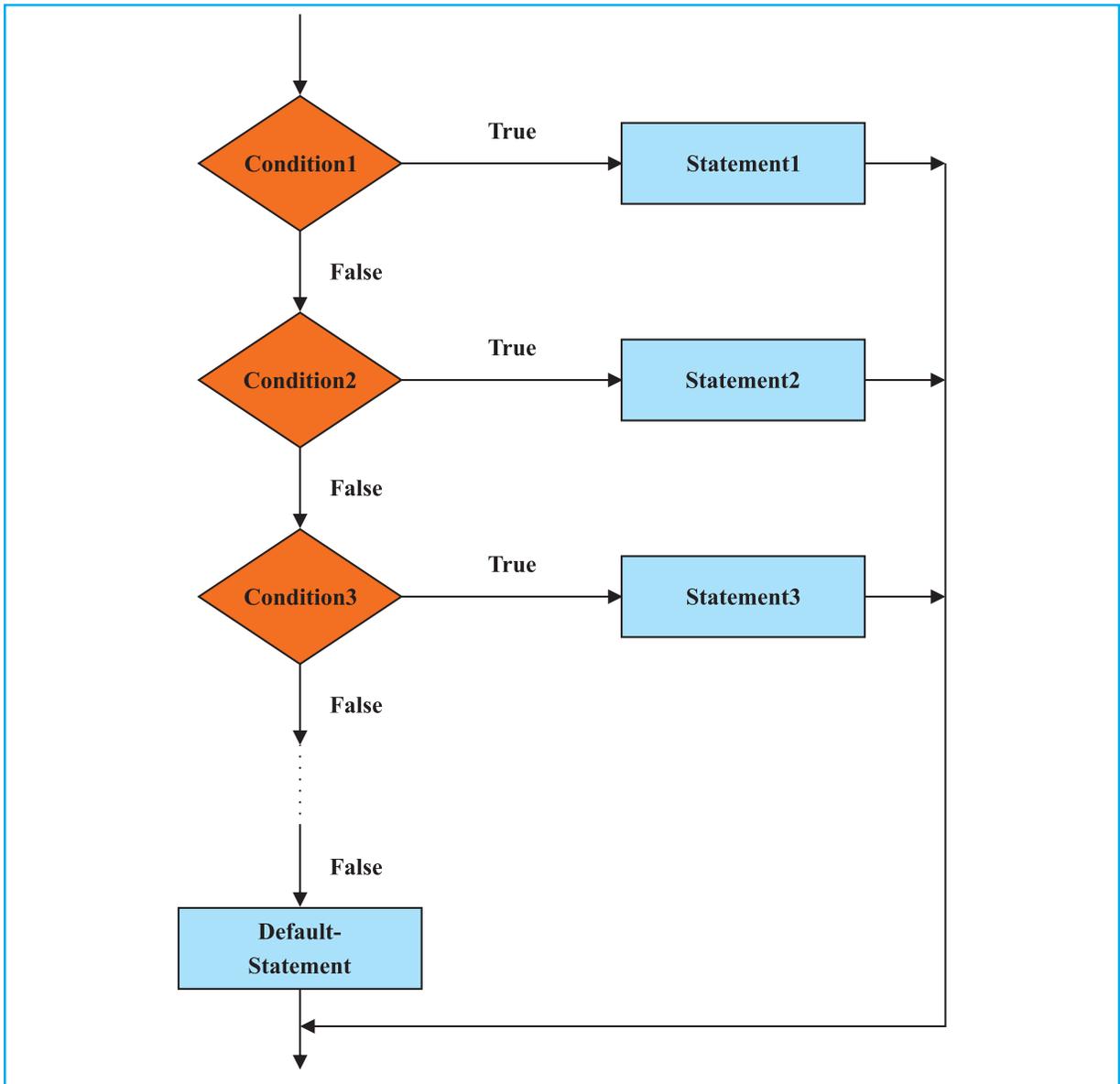


Figure 5.7 : Use of Multi-choice using decision

**Connector:** Small circle with single capital letter is used as connector. When a problem is relatively large and complex, flowchart is either complex or may not fit into a page. Connectors are used to connect two parts of flowchart within a page or in another page. Connector is always used in pair where circle having letter say “A” with inward arrow (coming out of some symbol) shows that flowchart continues in same part of same page or in another page denoting circle with same letter “A” but outward arrow connecting to the other part in same page or other page. Figure 5.8 shows the pair of connectors with letter “A”.

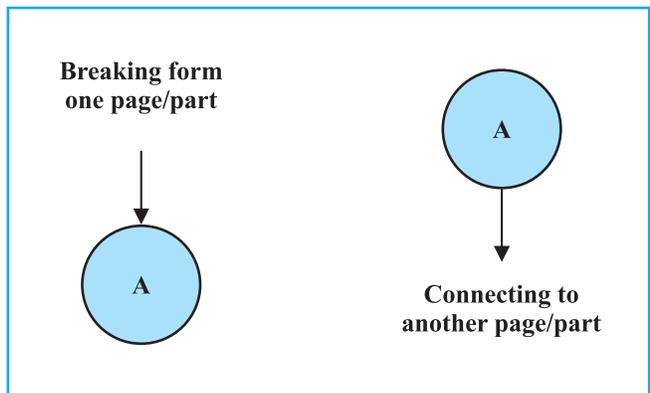


Figure 5.8 : Connectors

## Sequential and Decision-making Examples

Let us now use symbols learnt in previous section to develop the flowcharts for various problems including simple sequential steps and decision making.

**Problem 1 :** Draw a flowchart for adding two given numbers.

### Solution :

Problem is very simple and needs to receive two numbers, add them and finally produce the addition of them. Figure 5.9 shows the flowchart.

As shown in figure 5.9, flowchart starts with Start symbol. Then it receives two numbers using variables N1 and N2. Third step is the process step denoted by rectangle to add N1 and N2 into variable Sum. Forth step prints the addition stored in variable Sum. Last step is End to finish the flowchart.

We can easily draw the flowchart for the subtraction by making small changes in figure 5.9. First change is to replace the process step (third symbol) by

$$\text{Diff} = N1 - N2$$

and then changing the forth step with

output the difference

Print Diff

**Problem 2 :** Draw a flowchart to compute the simple interest using

$$I = (P \times R \times N) / 100$$

where P = Principal amount, R = Rate of interest and N = Period.

### Solution :

Let us take an example where P = 100000, R = 7% and N = 1 year.

Then interest  $I = (100000 \times 7 \times 1) / 100 = 7000$ .

Figure 5.10 shows the flowchart. As seen in the flowchart, three variables P, R and N are used to receive the input and then process box computes the interest I, using above mentioned formula and finally interest calculated is printed.

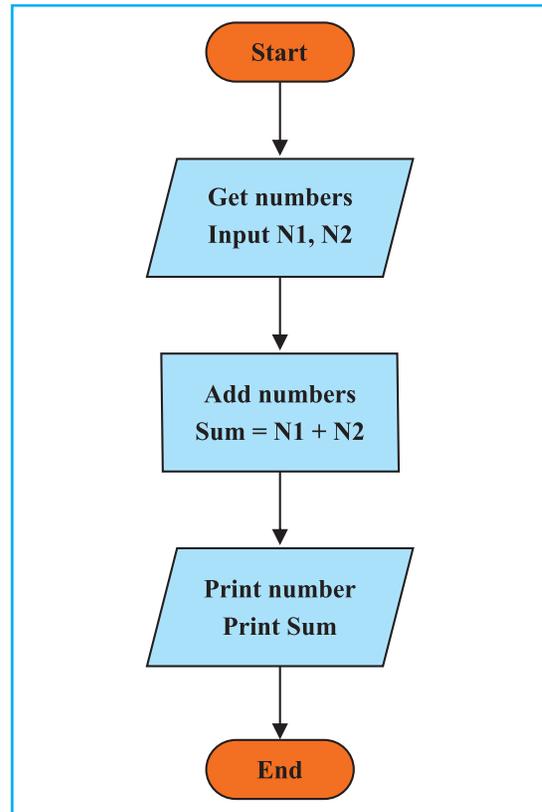


Figure 5.9 : Flowchart to add two numbers

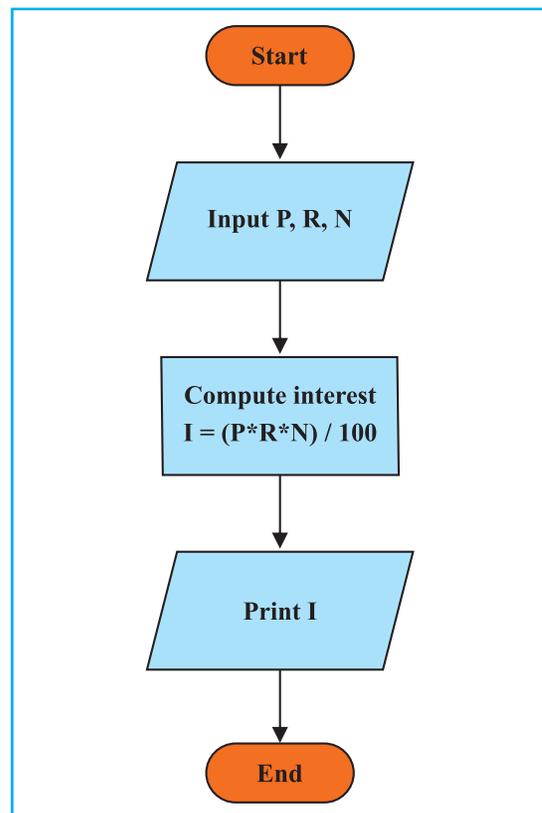


Figure 5.10 : Flowchart to compute simple interest

**Problem 3 :** Draw a flowchart to compute area and periphery of the circle.

**Solution :**

To compute, area and periphery of the circle, we need to accept radius as input. We can compute area using  $A = \pi R^2$  and the periphery using  $P = 2\pi R$ . Figure 5.11 shows the flowchart. Observe that we are computing two values in a single processing step. We can compute as many as value in a single process box if they are in sequence.

**Problem 4 :** Draw a flowchart to find the maximum of two numbers.

**Solution :**

This flowchart uses diamond symbol to compare two numbers using “>” (greater than) sign and then prints one of them whichever is larger. Figure 5.12 shows the flowchart.

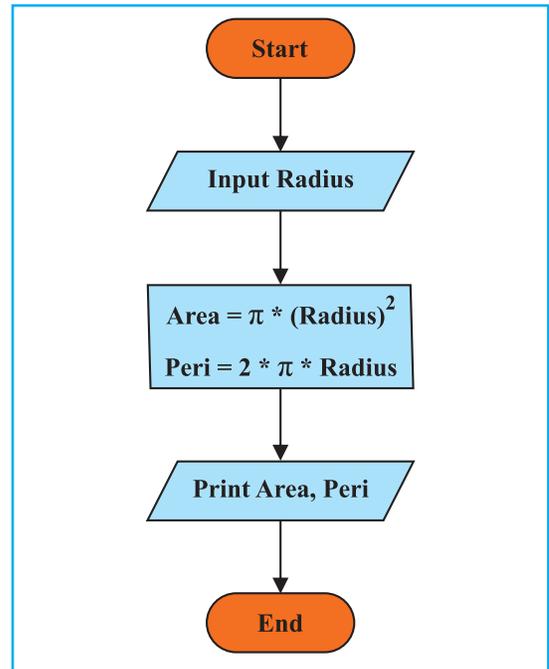


Figure 5.11 : Flowchart to compute Area and Periphery of Circle

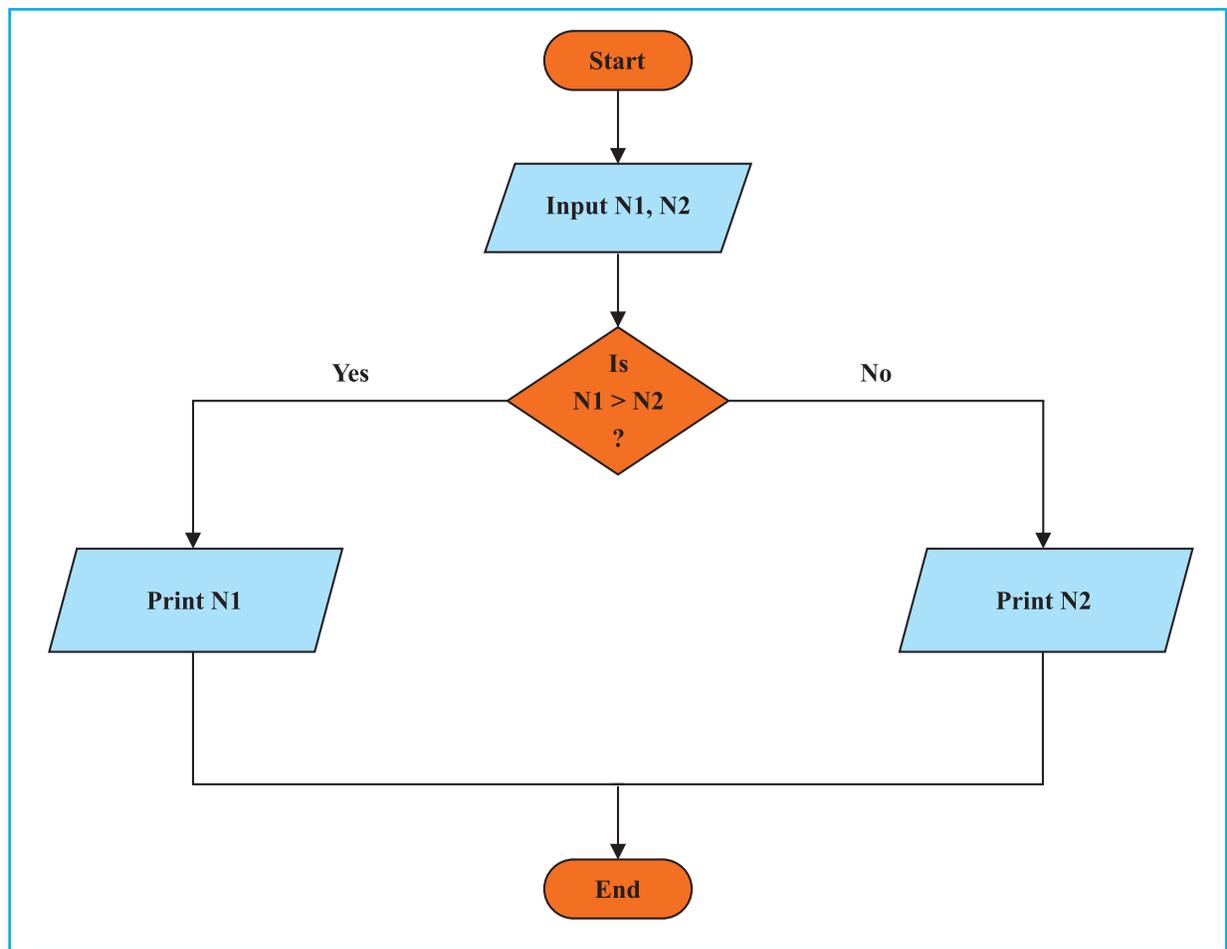


Figure 5.12 : Flowchart to find Maximum of Two Numbers

As shown in flowchart, after receiving two numbers N1 and N2, they are compared in decision box using condition  $N1 > N2$ . The decision box creates two paths, one if the condition is true (printing N1) and another if the condition is false (printing N2). After printing one of them flowchart ends.

If we simply change the condition to  $N1 < N2$  in decision box of the flowchart in figure 5.12, then it will print the minimum of N1 and N2.

**Problem 5 :** Draw a flowchart to find the maximum of two numbers and check whether the maximum number is odd or even.

**Solution :**

Figure 5.13 shows the flowchart for the problem. As seen in flowchart, first maximum number is found (refer figure 5.12) and stored in the variable Max. Then, we use decision box to test whether Max is odd or even using modulo by 2 (Modulo i.e. % operator gives the remainder after division). If the  $Max \% 2 = 0$ , then it is Even, otherwise ( $=1$ ) it is odd. Finally, we print the Max with label “Max is Even” or “Max is Odd”.

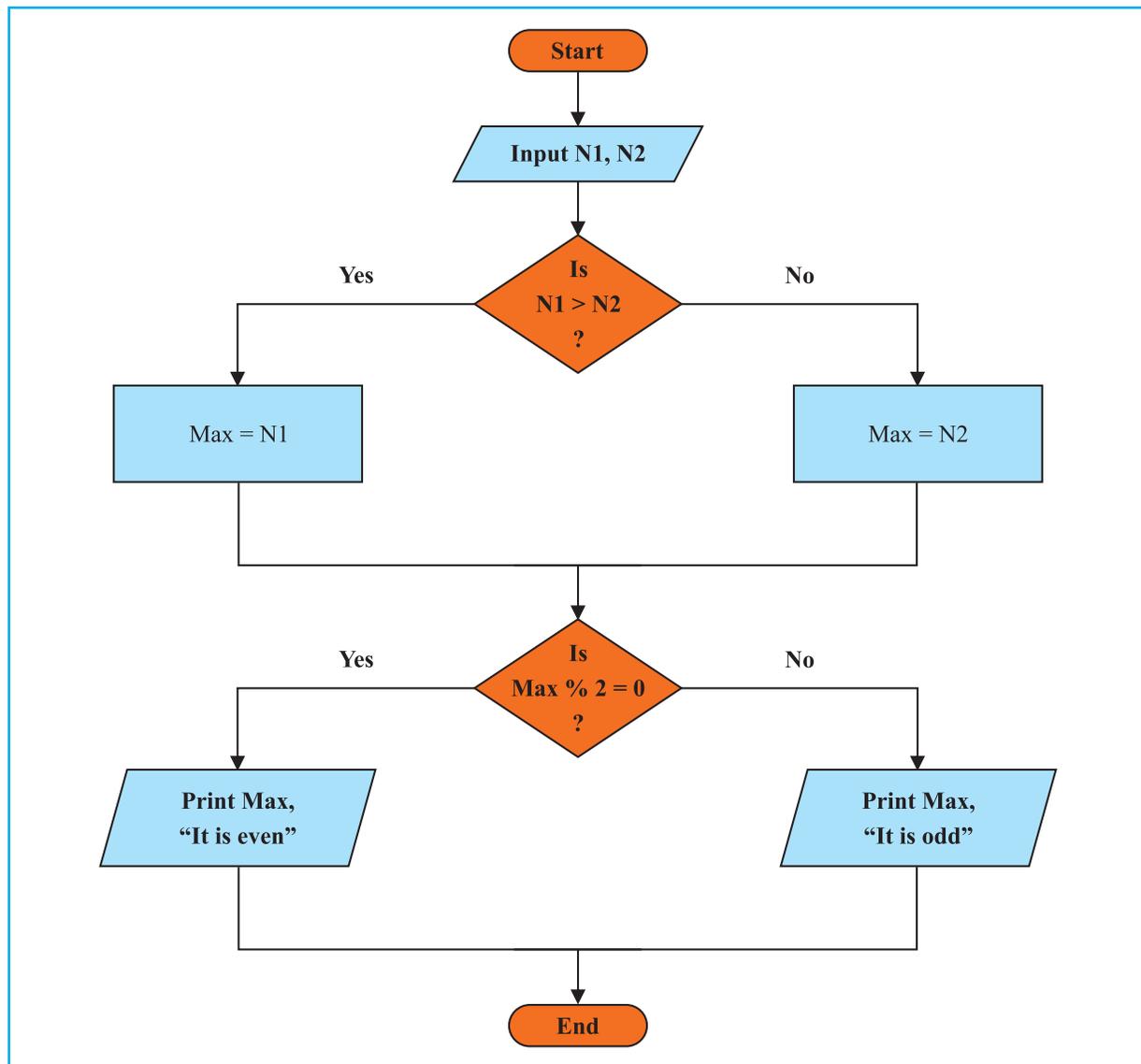


Figure 5.13 : Flowchart to find Maximum of Two is Odd or Even

## Nested Conditions and Looping with Examples

Nested condition means condition inside the condition. Sometimes, it is necessary that after applying one condition, we need to apply another condition based on the outcome of the first condition either true or false. Consider the situation shown in figure 5.14. First, we evaluate the Condition1. If it is true, then we apply Condition2 and continue further based on whether Condition2 is true or false. Same way, if Condition1 is false, we will apply Condition3 and continue further based on its outcome either true or false.

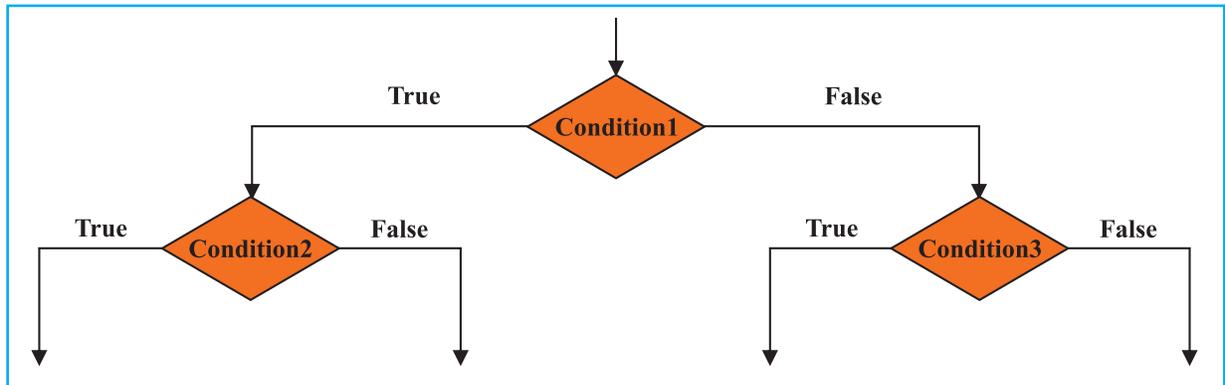


Figure 5.14 : Nested Conditions

**Problem 6 :** Draw a flowchart to find minimum of three numbers.

**Solution :**

Figure 5.15 shows the flowchart for finding the minimum of three numbers. Flowchart first accepts three numbers using variables N1, N2 and N3. Then decision box compares N1 and N2 ( $N1 < N2$ ). If N1 is less than N2, then nested decision box (left side) compares N1 with N3 ( $N1 < N3$ ) to find whether N1 is minimum or N3. If N2 is less than N1, then another nested condition box (right side) checks  $N2 < N3$  to find whether N2 is minimum or N3. Finally, the minimum of all three is printed.

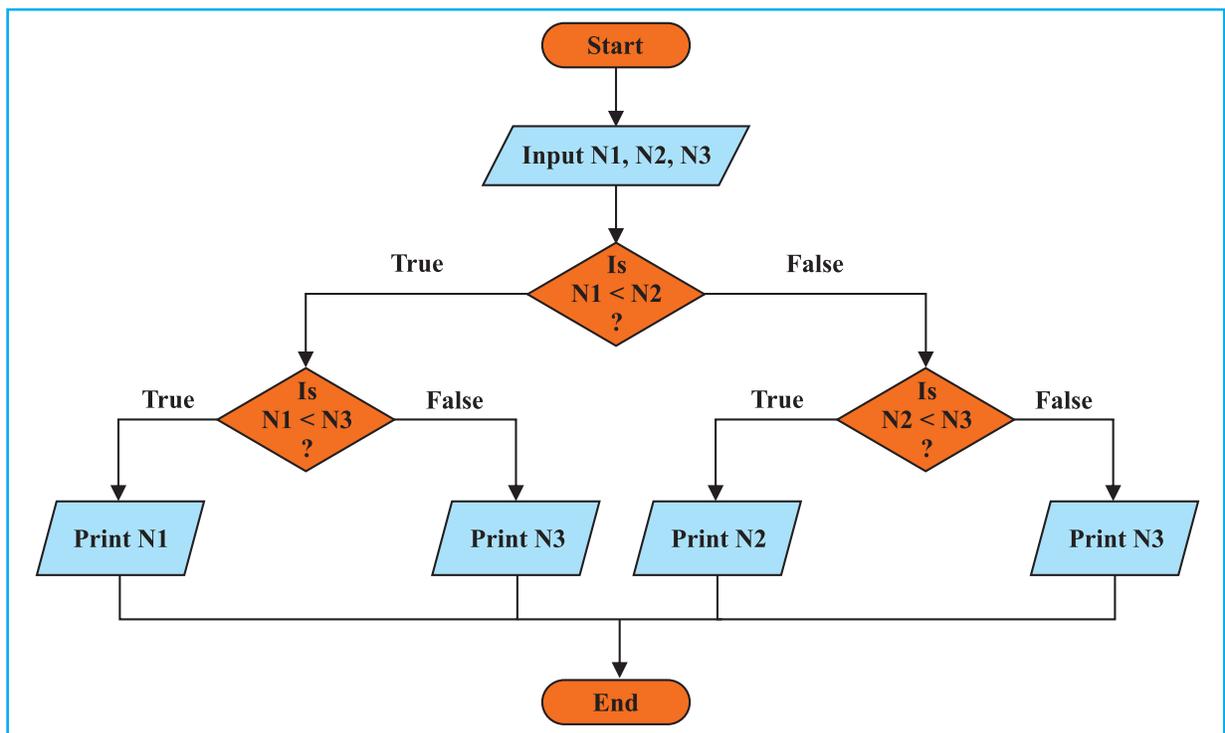


Figure 5.15 : Flowchart to Find Minimum of Three Numbers

For example, assume that  $N1 = 5$ ,  $N2 = 2$  and  $N3 = 7$ . Condition  $N1 < N2$  ( $5 < 2$ ) is false. Control moves to the nested condition on right hand side which checks  $N2 < N3$  ( $2 < 7$ ) which is true and finally minimum of three  $N2 = 2$  is printed. You can test it with various combinations of  $N1$ ,  $N2$  and  $N3$ .

It is also possible that a task needs to repeat (looping) some set of statements based on some condition. Earlier we have taken an example to add 1 to 10 numbers, in which we have to repeat statement to add next value to Sum and increment  $I$  by 1 every time as long as  $I$  is less than or equal to 10. Let us understand it through following example.

**Problem 7 :** Draw a flowchart to add 1 to 10.

**Solution :**

Flowchart is shown in figure 5.16. First process box initializes the  $Sum = 0$  and  $I = 1$ . Then decision making statement  $I \leq 10$  is performed and if it is true then process box containing  $Sum = Sum + I$  and  $I = I + 1$  is executed. After execution, the arrow points back to decision box and condition  $I \leq 10$  is again evaluated with updated value. It is repeated as long as  $I \leq 10$ . When  $I = 11$ , condition becomes false and control moves to the next step after loop and finish.

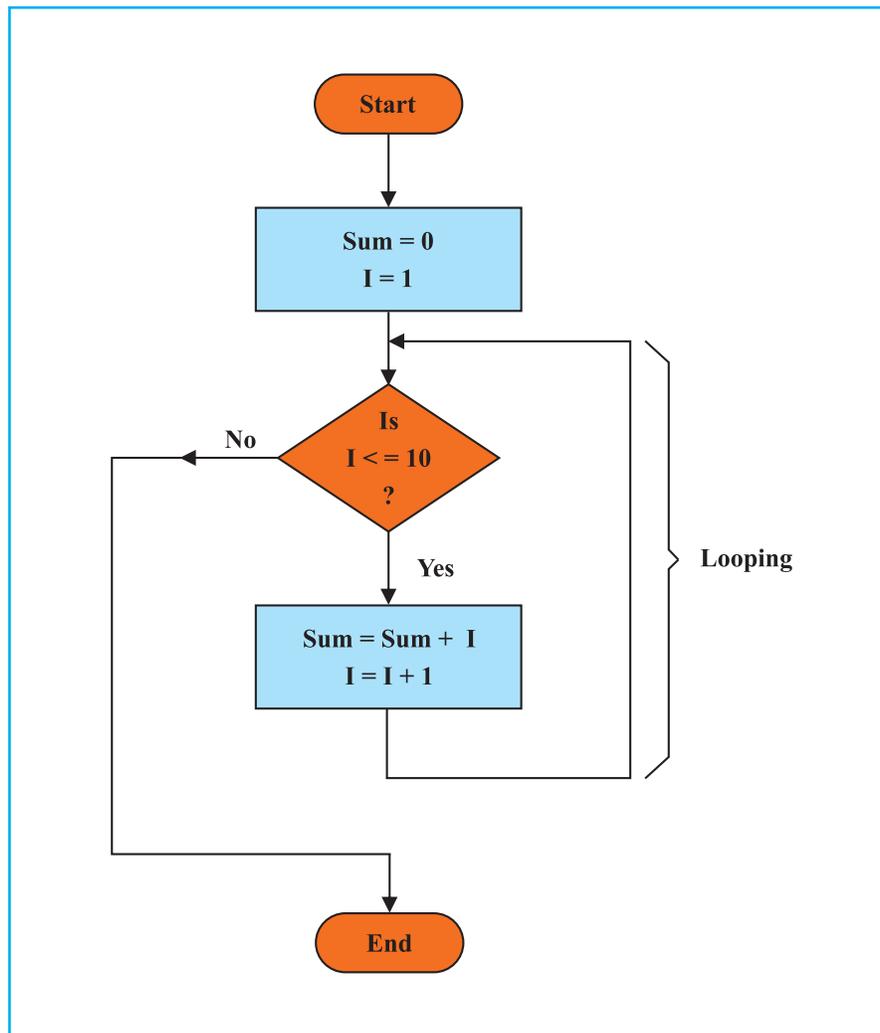


Figure 5.16 : Flowchart to Add 1 to 10

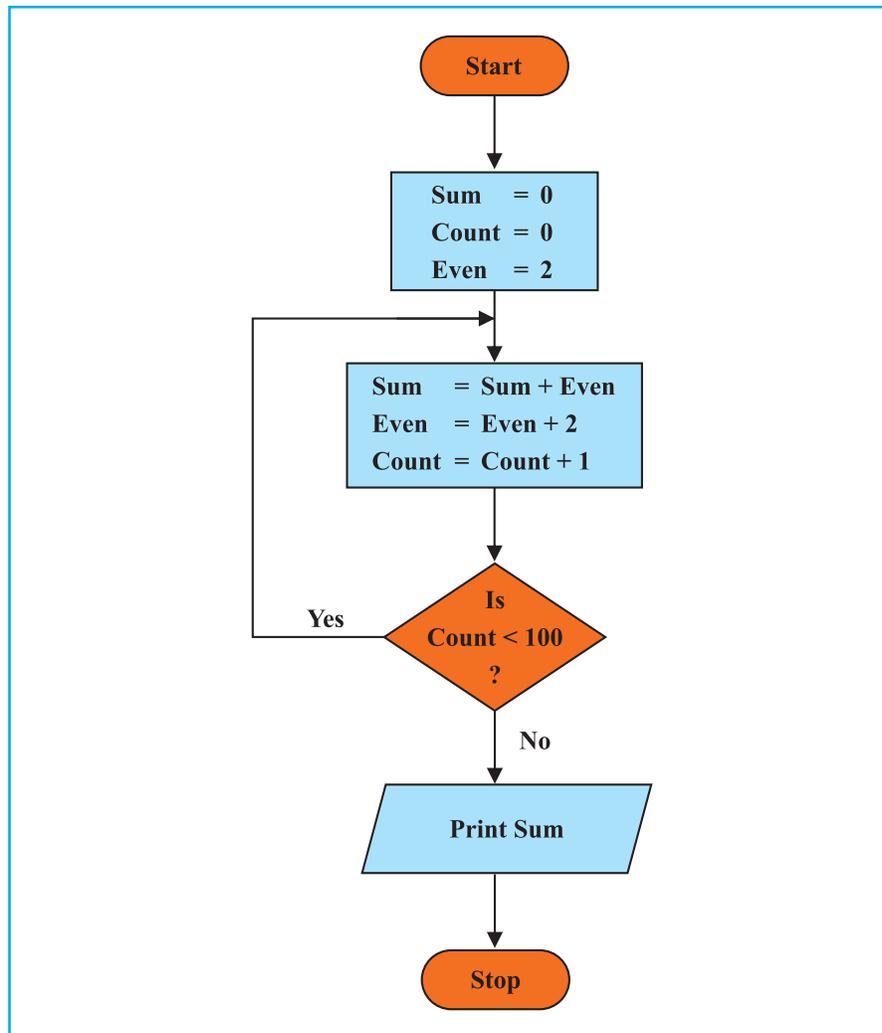
**Problem 8 :** Draw a flowchart to sum the first 100 even numbers.

**Solution :**

First 100 even numbers are covered in the range 1 to 200 and summing them means

$$2 + 4 + 6 + \dots + 200$$

Figure 5.17 shows the flowchart. We need three variables: Sum for storing addition of all even numbers, Count to keep track of when to finish and Even for storing next even number. They are initialized as Sum = 0, Count = 0 and Even = 2 (first even number). We will add next even number in Even to Sum and then increment Even by 2 and Count by 1. After processing it, flowchart checks if Count is less than 100, then we will repeat the process otherwise task is finished.



**Figure 5.17 : Flowchart to Add First 100 Even numbers**

### Overview of Algorithm

Algorithm is a stepwise solution of a given problem in logical order. It is just like a recipe of any food describing exactly what to do in specific order. Algorithm can be written using descriptive phrases or pseudo codes/instructions. Pseudo codes use words and symbols to represent operations or steps.



They are very compact and easy to understand. Algorithm gives the decomposition of problem into small steps which are easily understandable. Algorithm looks very similar to computer program except it uses common syntax like pseudo codes rather than specific syntax of a particular programming language. It is very easy to convert an algorithm to a computer program as whole solution is already developed using steps in a logical order. Writing algorithm before developing program ensures clarity and correctness of the program.

Following is an example of an algorithm to find the area of rectangle.

1. START
2. Input Length, Breadth
3. Compute Area = Length \* Breadth
4. Print Area
5. STOP

Observe that each step in an algorithm is given number starting from 1. Algorithm starts with START. The pseudo codes Input, Compute, Print denotes the input, computation and output operations respectively. The last step of the algorithm is always STOP denoting the end of the algorithm. Above algorithm first receives Length and Breadth, then it computes Area by multiplying Length and Breadth and finally prints the Area as an output.

## Examples of Algorithm

Let us write the various algorithms including sequence, decision making and repetition so that we can understand it in depth.

**Problem 9 :** Write an algorithm to find the square of a given number.

**Solution :** The algorithm is as follows.

1. START
2. Input N
3. Compute Square = N \* N
4. Print Square
5. STOP

It is very simple and follows the steps in sequence from 1 to 5. It accepts N, computes Square and then prints the Square.

Let us take the examples which includes the decision making and repetition/looping.

**Problem 10 :** Write an algorithm to find maximum of two numbers.

**Solution :** The algorithm is as follows.

1. START
2. Input N1, N2
3. If  $N1 > N2$ , goto 6
4. Print N2
5. goto 7



6. Print N1
7. STOP

First algorithm reads two numbers using variables N1 and N2. Step 3 to step 6 implements the decision making where step 3 compares N1 and N2 ( $N1 > N2$ ) and, if true, then it jumps to step 6 to print N1 followed by STOP. If it is false, then follows the step 4 to print N2 and step 5 sends control to step 7 to STOP. If we don't use goto 7 in step 5 then after printing N2, it will also print N1 which is wrong. Hence, care has been taken to avoid it. Flowchart for the same is given in figure 5.12. Compare the steps in algorithm and flowchart so that it will be very clear that how we can convert algorithm to flowchart and vice versa.

**Problem 11 :** Write an algorithm to add first 100 even numbers.

**Solution :** The algorithm is as follows.

1. START
2. Initialize Sum = 0
3. Initialize Count = 0
4. Initialize Even = 2
5. Compute Sum = Sum + Even
6. Compute Even = Even + 2
7. Increment Count by 1
8. If Count < 100, goto 5
9. Print Sum
10. STOP

Steps 2, 3 and 4 initializes the variables Sum, Count and Even respectively. Step 5 to step 8 implements the repetition or loop. Step 5 adds current Even value to Sum. Step 6 adds 2 to the Even to get next even number. Step 7 increments Count by 1 as already one even number is added to the Sum. Step 8 checks if Count is less than 100. If it is, then it jumps to step 5 and loop is repeated again. When Count = 100, condition is false and it follows step 9 to print the Sum. Thus, step 5 to step 8 are repeated 100 times and each time next even number (2,4,6, ..., 200) is added to Sum and Count goes from 0 to 99 and when it is 100, the loop is over. Flowchart for the same is given in figure 5.17. Compare it with above algorithm to understand how a loop is implemented.

### Flowchart Vs. Algorithm

We have studied the flowcharts and algorithms with examples and understood that how sequence, decision and looping can be implemented which are most important components of problem solving. Any problem in real life is different combinations of these three elements. We can use either flowchart or algorithm or both to represent the process i.e. problem solution in structured and logical order step-by-step depending on the need or choice. Although both flowchart and algorithm are used as problem solving methods, they have their own merits and demerits. Let us compare flowchart and algorithm. It is given in following table.



<b>Flowchart</b>	<b>Algorithm</b>
Provides visual representation of process using symbols.	Uses pseudo codes / instructions to represent the process.
All the paths are visible and easy to follow.	Paths are hidden and needs to read the steps to understand.
Shows the flow of the process which is good for visual learners.	Provides logic of the process using steps which is good for logical thinkers.
Good for planning and explaining the process.	Good for writing computer programs as algorithm is very similar to computer program.
Good for beginners.	Good for computer programmers.

## Summary

We started this chapter with the introduction to problem solving and its importance. The problem solving is a logical and step-by-step process to find the solution of the problem. Problem solving is at the heart of the Computer Science and it is foundational skill for the Computer Science as well as for the day to day life. We have studied two widely used methods for problem solving which uses structured approach to solve the problems. They are flowcharts and algorithms. Flowchart is a pictorial representation of the solution using visual diagrams to show the flow of the solution. We have developed the flowcharts for many small and simple problems for better understanding. Examples also provided opportunities to trace the flowchart to follow different paths based on decisions. Algorithm provides the step-by-step solution of the problem. For better understanding and clarity, chapter provides examples of writing algorithms for the same problems used to develop the flowcharts. Chapter ends with the comparison between both the problem-solving methods i.e. flowchart and algorithm. These methods reinforce logical thinking and clarity on expressing the solutions. It also allows students to test the solution by providing various inputs and following paths based on decisions. This chapter laid down the strong foundation for learning programming with C in coming chapters.

## EXERCISE

1. What is problem solving? Give an example.
2. Mention the problem-solving methods with their benefits.
3. What is flowchart? List the symbols used by flowchart with their function.
4. What types of operations are normally part of process box of the flowchart?
5. Which symbol is used for decision making? Explain its use with an example.
6. How would you create a looping using flowchart symbols?
7. What is algorithm? Give a simple example.
8. Give an example of repetition using algorithm.



9. Give the drawbacks of algorithm.  
10. Compare flowchart and algorithm.

**11. State whether true or false.**

- (1) Arrow is used to connect two steps in flowchart.
- (2) All the possible paths are visible in algorithm.
- (3) Problem solving is logical and step-by-step process to solve the problem.
- (4) Multi-choice can be implemented by diamond.
- (5) Flowchart always contains Start and End steps.

**12. Fill-in the blanks.**

- (1) ..... is pictorial representation of the solution.
- (2) Algorithms use ..... code.
- (3) Connectors use ..... symbol in flowchart.
- (4) Normally arithmetic and logical operations are placed in ..... box.
- (5) ..... and ..... are structured methods for problem solving.

**13. Multi-choice questions. Choose the most correct answer.**

- (1) Which of the following symbol is used for Start and End of a flowchart?  
(a) Rectangle      (b) Diamond      (c) Oval      (d) Circle
- (2) Which of the following provides visual representation of problem solution?  
(a) Algorithm      (b) Flowchart  
(c) Program      (d) Process
- (3) Which of the following symbol is used to denote the process in flowchart?  
(a) Rectangle      (b) Diamond      (c) Oval      (d) Circle
- (4) Which of the following is used to distinguish connector pairs?  
(a) Direction of arrows      (b) Letter in circle  
(c) Size of the circle      (d) Colour of the circle
- (5) Comparison is used in which of the following?  
(a) Process      (b) Decision-making  
(c) Input      (d) Output
- (6) Which of the following is best for identifying different paths within solution?  
(a) Flowchart      (b) Algorithm  
(c) Flowchart and Algorithm both      (d) We can not see the path
- (7) Which of the following uses pseudo codes to represent steps of solution?  
(a) Program      (b) Flowchart      (c) Algorithm      (d) Process

- (8) Which of the following is normally used to perform computations?
- (a) Process box (b) Decision box  
(c) Input box (d) Output box
- (9) Which of the following is used when flowchart is split into more than one pages?
- (a) Letter with arrows (b) Connectors  
(c) Arrows (d) Special symbols
- (10) When we need to compute two values, we can
- (a) Put them in one process box one by one  
(b) Put them in two separate process box one after another  
(c) Both A and B  
(d) Must be in two separate box one after another

### Laboratory Exercise

1. Draw a flowchart to convert Fahrenheit to Celsius using  $C = (5.0/9.0) * (F - 32)$ .
2. Draw a flowchart to check whether given positive number is odd or even.
3. Draw a flowchart to add only odd numbers from 1 to 100.
4. Draw a flowchart to accept the students mark of a subject from 100 and print “Fail”, if marks is  $\leq 35$ , “Second class” if marks is  $> 35$  and  $\leq 60$ , print “First class” if marks  $> 60$  and  $\leq 70$ , otherwise print “Distinction”.
5. Draw a flowchart to guess a number. It accepts number repeatedly from user until number is 7. When number is 7, it prints “Congratulations!” and finish.
6. Draw a flowchart to find minimum from given 10 numbers.
7. Write an algorithm to find sum and difference of two given numbers
8. Write an algorithm to find maximum of three numbers.
9. Write an algorithm to add only even numbers from given N numbers as input.
10. Write an algorithm to add following series :  $1-2+3-4+5-6+ \dots \dots \dots N$ .

