Experiment No. 1

Introduction to Algorithm and Flowcharts

Objective:
The objective of this lab is to understand the concepts of algorithms and flowcharts and how they are used to design and represent a program's logic before coding in the C programming language.

Learning Outcome:
Upon completion of this experiment, the students shall be able to:

1. Define algorithms and create step-by-step procedures for problem-solving.
2. Use flowchart symbols to visually represent algorithms and logical sequences.
3. Analyze problems, develop logical solutions, and effectively test/debug C programs.

Tools Required:

- Pen and Paper (for manual drawing of flowcharts)
- Software for creating flowcharts (optional, e.g., Lucidchart, draw.io)

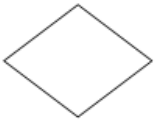
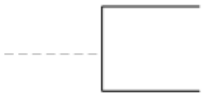
Introduction:
Algorithms and flowcharts serve as foundational tools for effective problem-solving and program development. An algorithm is a step-by-step procedure or formula for solving a problem, offering programmers a clear and concise method to follow. It breaks down complex problems into manageable tasks, ensuring each step is logically sound and sequentially correct. A flowchart, on the other hand, is a visual representation of an algorithm. It uses standardized symbols to depict the flow of control and data through the program, making it easier to understand and communicate the structure of a program. Together, algorithms and flowcharts help programmers plan, visualize, and troubleshoot their code, fostering a deeper understanding of programming logic and improving the overall efficiency and reliability of the software development process.

Properties of Algorithm:

1. **Finiteness:** An algorithm must always terminate after a finite number of steps. This means that after every step, one reaches closer to the problem's solution, and after a finite number of steps, the algorithm reaches an end point.
2. **Definiteness**: Each step of an algorithm must be precisely defined. This is done by well-thought-out actions that must be performed at each step of the algorithm. Also, the actions are defined unambiguously for each activity in the algorithm.
3. **Input:** Any operation you perform needs some beginning value/quantities associated with different operations activities. So the value/quantities are given to the algorithm before it begins.
4. **Output:** One always expects output/result (expected value/quantities) in terms of output from an algorithm. The result may be obtained at different stages of the algorithm. If some result is from the intermediate stage of the operation, then it is known as intermediate result, and result obtained at the end of the algorithm is known as end result. The output is the expected value/quantities that always have a specified relation to the inputs.

5. **Effectiveness:** Algorithms to be developed/written using basic operations. Actually, operations should be basic, so that even they can in principle be done exactly and in a finite amount of time by a person, using paper and pencil only.

Shapes and Symbols Used in Flowcharts:

Flowchart Symbol	Symbol Name	Description
	Terminal (Start or Stop)	Terminals (Oval shapes) are used to represent start and stop of the flowchart.
	Flow Lines or Arrow	Flow lines are used to connect symbols used in flowchart and indicate direction of flow.
	Input / Output	Parallelograms are used to read input data and output or display information
	Process	Rectangles are generally used to represent process. For example, Arithmetic operations, Data movement etc.
	Decision	Diamond shapes are generally used to check any condition or take decision for which there are two answers, they are, yes (true) or no (false).
	Connector	It is used connect or join flow lines.
	Annotation	It is used to provide additional information about another flowchart symbol in the form of comments or remarks.

Steps to Create an Algorithm and Flowchart:

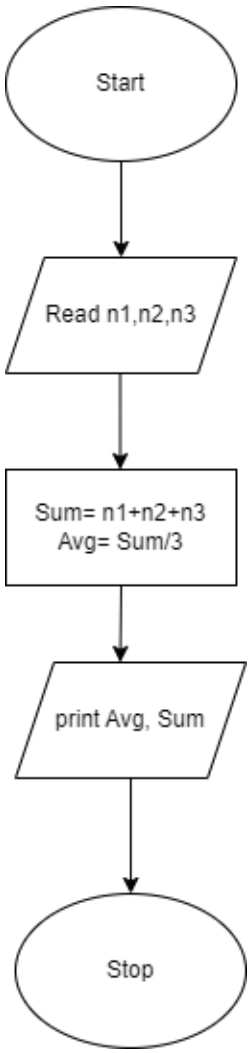
1. **Define the problem**: Understand the problem statement clearly.
2. **Develop an algorithm**: Write down the step-by-step instructions to solve the problem.
3. **Draw the flowchart**: Represent the algorithm using flowchart symbols and connect them with arrows.

Example Problem 1: Write an algorithm and flowchart to find the sum and average of three numbers.

Flowchart:

Algorithm:

Step 1: Start
Step 2: Declare variables num1, num2, num3, sum, and average.
Step 3: Read values num1, num2, and num3.
Step 4: Add num1, num2, num3, and assign the result to sum.
sum←num1+num2 +num3
average ← sum/3
Step 5: Display the sum and average
Step 6: Stop



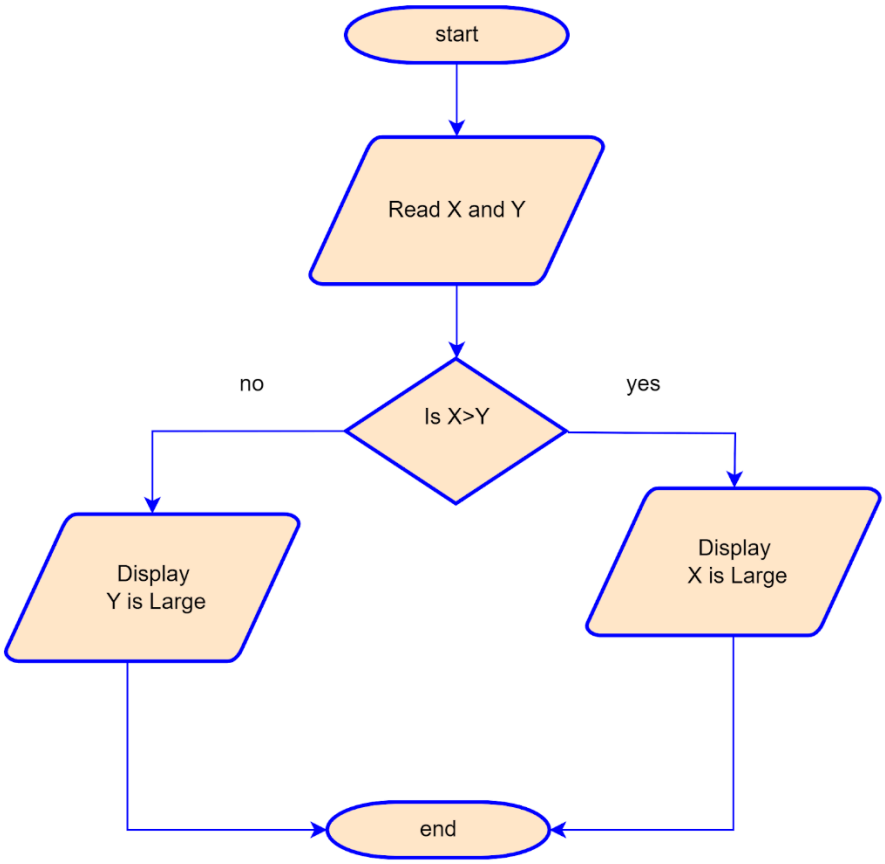
The ***branch*** refers to a binary decision based on some condition. If the condition is true, one of the two branches is explored; if the condition is false, the other alternative is taken. This is usually represented by the 'if-then' construct in pseudo-codes and programs. In flowcharts, this is represented by the diamond-shaped decision box. This structure is also known as the selection structure.

Example Problem 2: Write an algorithm and Flowchart to Find the Largest of Two Numbers

Algorithm:

Step 1: Start
Step 2: Read X, Y
Step 3: Is X>Y
Print X is large
Else Print Y is large.
Step 4: Stop

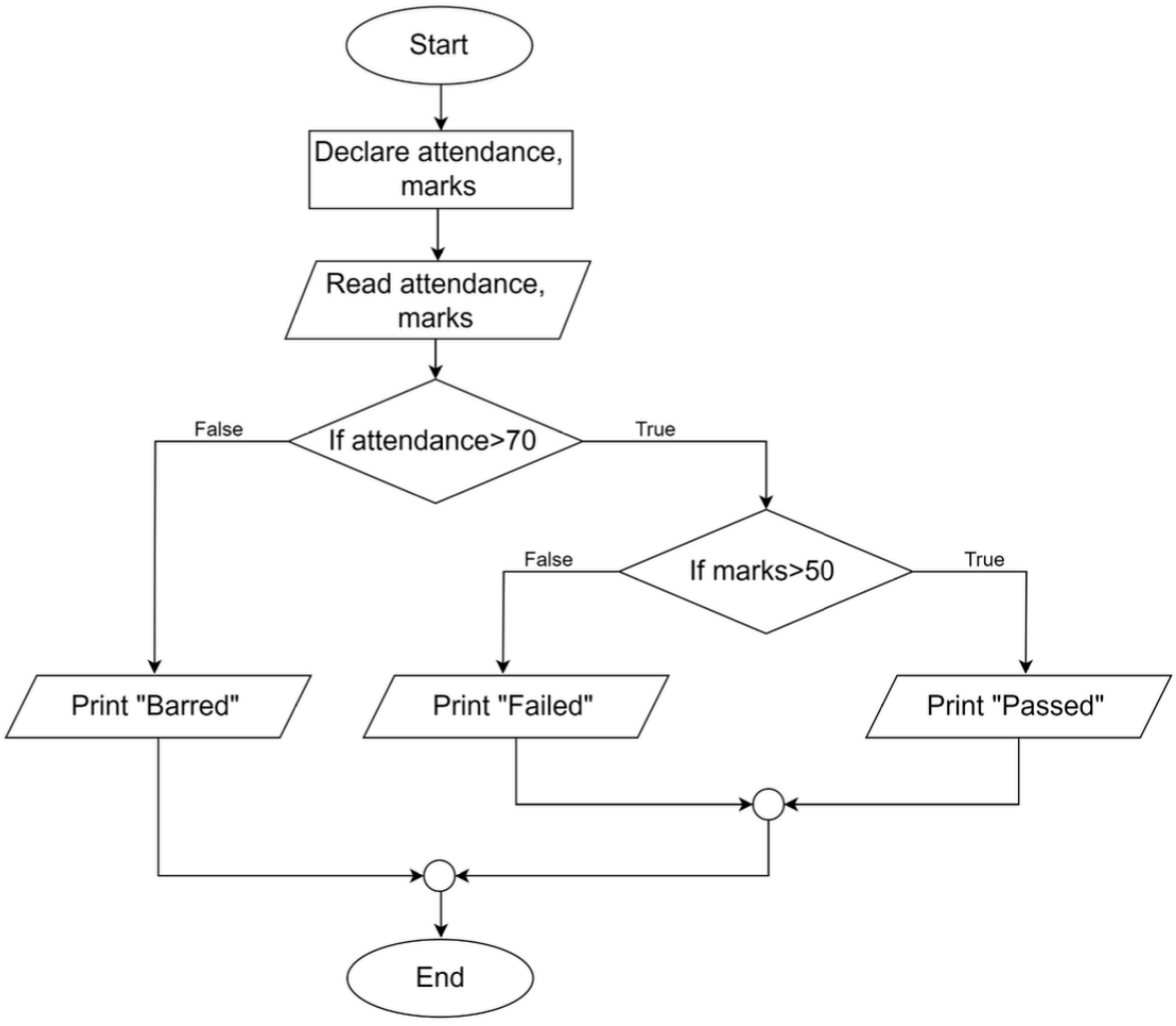
Flowchart:



Example Problem 3: Suppose you are tasked to submit grades of EEE-103. You will ask for a total of 2 numbers: the attendance percentage and the total obtained marks. If the attendance percentage is greater than 70% and the student has obtained more than 50 marks then the program will print "Passed" otherwise "Failed". If attendance is less than 70% no need to check the marks and print "Barred".

Draw a flowchart for the above problem.

Flowchart:



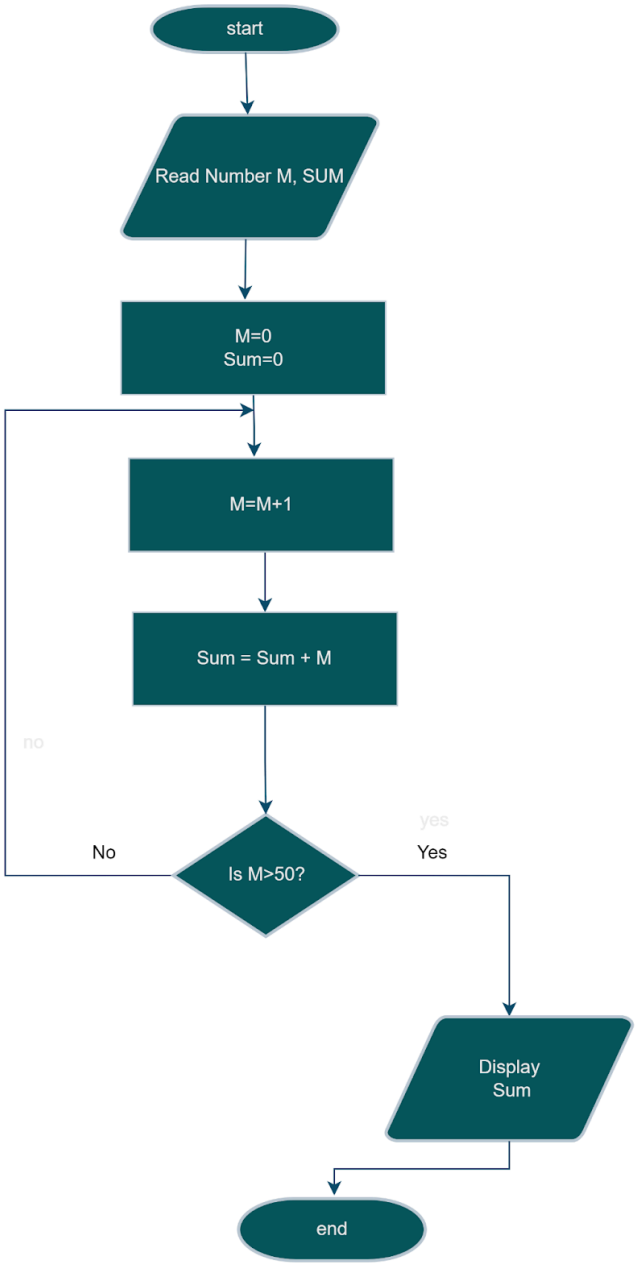
Loop: The *loop* allows a statement or a sequence of statements to be repeatedly executed based on some loop condition. It is represented by the 'while' and 'for' constructs in most programming languages. In the flowcharts, a back arrow hints at the presence of a loop. A trip around the loop is known as iteration. You must ensure that the condition for the termination of the looping must be satisfied after some finite number of iterations, otherwise, it ends up as an infinite loop, a common mistake made by inexperienced programmers. The loop is also known as the repetition structure.

Example Problem 4: Write an algorithm and Flowchart to calculate the Sum of The First 50 Numbers

Algorithm:

Step 1: Read number M and Sum
Step 2: Declare number M= 0 and Sum= 0
Step 3: Determine M by M= M+1
Step 4: Calculate the sum by the formula: Sum= Sum + M
Step 5: Add a loop between steps 3 and 4 until M= 50.
Step 6: Print Sum

Flowchart:

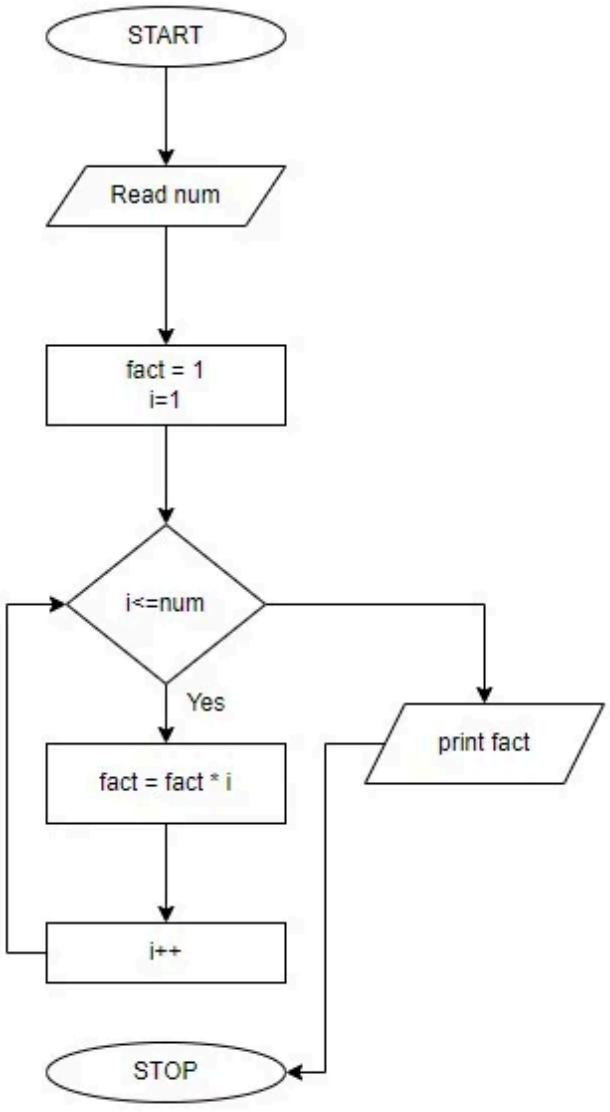


Example Problem 5: Write an algorithm and draw the flowchart to find the factorial of a number.

Algorithm:

Step 1: Start
Step 2: Read a number: num
Step 2: Initialize variables: i = 1, fact = 1
Step 3: if i <= n go to step 4, otherwise go to step 6
Step 4: Calculate: fact = fact * i
Step 5: Increment the i by 1 (i=i+1) and go to step 3
Step 6: Print fact
Step 7: Stop

Flowchart:



Lab report:

1. Draw a flowchart for a program that calculates and prints the grade for a student based on their score. First ask the user for the obtained marks. Determine the grade based on the following grading scale:

90 or above: A
80 - 89: B
70 - 79: C
60 - 69: D
Below 60: F

Print the grade.

2. Write an algorithm and draw the flowchart of a program that reads the radius of a circle and prints its circumference and area. If the area is larger than 100 print “Too large”, if the area is smaller than 50 print “Too small”; otherwise print “Perfect”.

3. Write an algorithm and draw the flowchart of a program that reads five numbers as input from the user, prints whether the numbers are odd or even and prints the average of the numbers.

4. Write an algorithm and flowchart of the following program: You are tasked with writing a program that calculates the total bill for a customer's purchase. The program should perform the following steps: Prompt the user to enter the prices of three items (item1, item2, and item3). Calculate the subtotal and add a 10% VAT to the total. If the total bill (including VAT) is more than 500, apply a 50 TK discount.

Print the total bill after applying the VAT and discount.