

The objective of this lab is to:

To understand and apply conditional statements and loops for solving basic programming problems.

ALERT!

1. This is an **individual lab**. You are **strictly NOT** allowed to collaborate with others, share screens, or communicate answers in any form.
2. **Use of AI tools (e.g., ChatGPT, Copilot, etc.) is strictly prohibited.** Any AI-generated content will be treated as academic dishonesty.
3. **Anyone caught in act of cheating would be awarded an “F” grade in this Lab.**

Task 01:

[5 marks]

Coordinate Quadrant Identification

Problem Statement 1:

Write a C++ program to accept a two digit number as coordinate point in an XY coordinate system and then separate XY points and determine in which quadrant the coordinate point lies.

Example:

Input:

Enter a Two Digit number as XY coordinate points: 79

Output:

The coordinate (7,9) lies in first Quadrant.

Multiplication without * operator:

Problem Statement 2:

Write a C++ program that takes **two integers** as input from the user and computes their product **without using the * operator**. Instead, use the concept of **repeated addition** or **loops** to perform the multiplication. The program should handle both positive and negative numbers correctly.

Example:

Input:

Enter the first number: 6

Enter the second number: 4

Output:

The product of 6 and 4 is: 24

Input:

Enter the first number: -3

Enter the second number: 5

Output:

The product of 6 and 4 is: -15

Task 02:

[5 Marks]

Processing Positive Integers with Conditions and Loops:

Problem Statement

Write a C++ program that continuously accepts **positive integers** from the user. The program should keep track of the **count, maximum, minimum, and average** of the entered numbers. The process should continue until the user enters **-1**, which will act as a **termination signal**. Upon termination, the program should display the calculated results.

Example:

Input:

Enter a Positive Number and -1 to terminate: 5
Enter a Positive Number and -1 to terminate: 7
Enter a Positive Number and -1 to terminate: 3
Enter a Positive Number and -1 to terminate: 9
Enter a Positive Number and -1 to terminate: -1

Output:

Number of positive integers is: 4
The maximum value is: 9
The minimum value is: 3
The average is: 6.00

Task 03:

[5 Marks]

Generate a Secret PIN

Problem Statement:

A bank generates a secret PIN for customers by taking their account number and:

- **Reversing the digits.**
- **Adding all the digits together.**
- **Concatenating both results.**

Write a program using a loop that takes an account number and generates the secret PIN.

Example:

Input:

Account Number = 4521
Step 1: Reverse = 1254
Step 2: Sum of digits = 12

Output:

Secret PIN = 125412

Task 04:

[10 Marks]

Problem Statement

Write a C++ program to display the following pattern of numbers in the form of a 5×5 matrix:

```
1 0 0 0 2
0 3 0 4 0
0 0 5 0 0
0 6 0 7 0
8 0 0 0 9
```

Instructions:

- You are required to use nested loops to generate the pattern.
- Do not hardcode all the numbers directly; use logic to place the values in their correct positions.
- Zeros should be printed where no number exists.
- Output should exactly match the format of the pattern.

Task 05:

[5 Marks]

Problem Statement

Write a C++ program that takes a single time input in 24-hour format (military time) and converts it into 12-hour format with AM/PM.

Input:

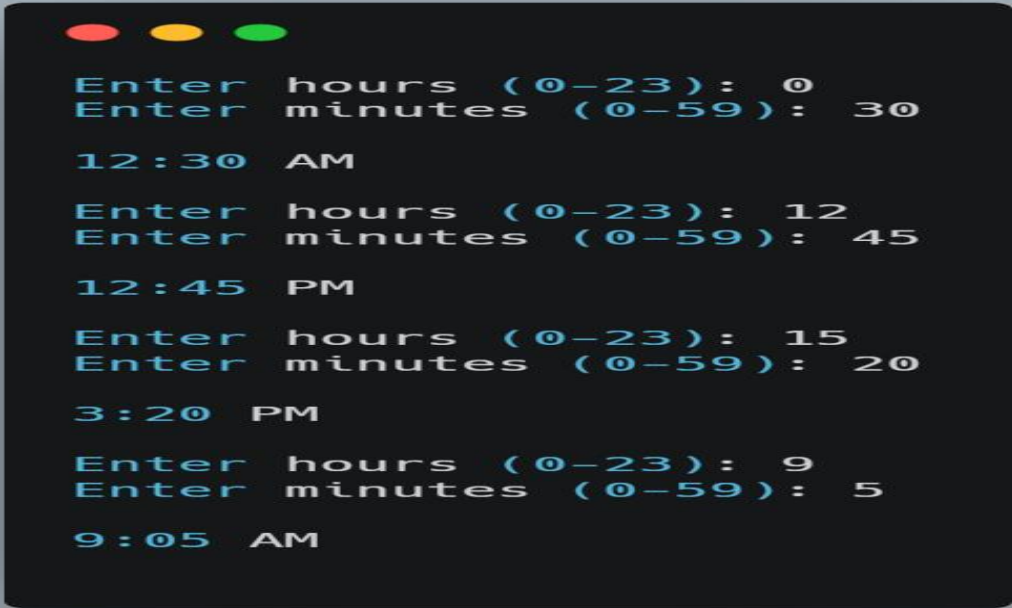
- Hours (0–23)
- Minutes (0–59)

Logic:

- If hours == 0 → 12 AM
- If $1 \leq \text{hours} < 12$ → hours AM
- If hours == 12 → 12 PM
- If hours > 12 → (hours - 12) PM

Output:

Display the time in 12-hour format with AM/PM.



```
Enter hours (0-23): 0
Enter minutes (0-59): 30
12:30 AM

Enter hours (0-23): 12
Enter minutes (0-59): 45
12:45 PM

Enter hours (0-23): 15
Enter minutes (0-59): 20
3:20 PM

Enter hours (0-23): 9
Enter minutes (0-59): 5
9:05 AM
```

Task 06: [10 Marks]

Armstrong Number Check

Problem Statement:

An Armstrong Number (narcissistic number) is a number that is equal to the sum of its own digits raised to the power of the number of digits.

Illustration:

- $153 \rightarrow 1^3 + 5^3 + 3^3 = 153$ (Armstrong)
- $9474 \rightarrow 9^4 + 4^4 + 7^4 + 4^4 = 9474$ (Armstrong)
- $123 \rightarrow$ Not Armstrong

Design an algorithm using a while loop to check if a number is Armstrong.

Example:

Input:

Enter a number: 153

Output:

153 is an Armstrong Number.

Input:

Enter a number: 123

Output:

123 is NOT an Armstrong Number.

Task 07:

[10 Marks]

Spiral Matrix display

Problem Statement:

Write a C++ program to fill an $n \times n$ square matrix with numbers from 1 to $n \times n$ in a spiral order (moving clockwise), and then display the matrix.

```
1 2 3 4 5
16 17 18 19 6
15 24 25 20 7
14 23 22 21 8
13 12 11 10 9
```

Instruction:

- You must use nested loops and appropriate logic to fill the matrix.
- The numbers should be placed in increasing order starting from 1 up to $n \times n$.
- The filling order should follow a spiral pattern: right \rightarrow down \rightarrow left \rightarrow up \rightarrow repeat until the matrix is filled.
- Finally, display the matrix in a formatted way.