Lab Experiment 01 - C Language

# Objective:

* To gain practical experience with advanced pointer concepts in C, including pointer arithmetic, pointers and arrays, and function pointers.

# Materials needed:

* Computer with a C compiler (e.g., GCC)
* Text editor or IDE

# Part 0: Quick Revision Exercises

**Use this template code for your work:** [**template\_code\_Part0.c**](https://drive.google.com/file/d/1nVHzgol6ugmq-3TnS6vF2k9u6MT94ivd/view?usp=sharing)

## Task 0.1: Basic Syntax and Data Types

* Declare variables of type int, float, double, char.
* Print their values and sizes using the sizeof operator.
* Demonstrate type casting (e.g., int → float, float → int).

## Task 0.2 Operators and Expressions

* Write a program that takes two integers as input.
* Perform all arithmetic operations: +, -, \*, /, %.
* Extend into a simple calculator using switch statement:
  + User chooses the operation symbol (+ - \* / %)
  + Program executes the selected operation.

## Task 0.3 Control Structures

* **Fibonacci Sequence:**
  + Print the first n terms of the Fibonacci sequence using a for loop (n is user input).
* **Guessing Game:**
  + Computer generates a random number between 1–100.
  + User repeatedly guesses until correct.
  + Program responds with "Too High" or "Too Low" hints.

## Task 0.4 Functions

* Write a function isPrime(int n) that returns 1 if n is prime, otherwise 0.
* Use it to print all prime numbers between 1–100.
* Write a recursive function factorial(int n) that calculates factorial.

## Task 0.5 Arrays and Strings

* **Reverse a string:**
  + Write a function to reverse a string without using library functions.
* **Find the 2nd Largest Element in an Array:**
  + Write a function that scans an integer array and prints the second largest element.

## Task 0.5 Arrays and Strings

* **Reverse a string:**
  + Write a function to reverse a string without using library functions.
* **Find the 2nd Largest Element in an Array:**
  + Write a function that scans an integer array and prints the second largest element.

## Task 0.6: File I/O Basics

* Write a program that:
  + Reads 5 integers from the user and stores them in a file (numbers.txt).
  + Reads back the integers from the file and prints them on the console.
* (Optional) Extend: Write results of prime-checker from Task 0.4 into a file.

## Task 0.7: File I/O Basics

* Write a program that demonstrates:
  + AND &, OR |, XOR ^, NOT ~, and bit-shifting <<, >>.
  + Example: Given x = 5, y = 9, show results of x & y, x | y, etc.
* Write a function that checks if a number is power of 2 using bitwise operators only.

## Task 0.8: Enumerations

* Define an ***enum Weekday { MON, TUE, WED, THU, FRI, SAT, SUN };***
* Write a program that takes a number (1–7) as input and prints the corresponding weekday.

## Task 0.9: Structures (Intro)

* Define a struct Point { int x; int y; };
  + Write a program that takes two points and calculates the Euclidean distance between them.
  + Write a function that checks if a number is power of 2 using bitwise operators only.

## Task 0.10: Command Line Arguments

* Write a program that accepts two integers from the command line and prints their sum.
  + Example: ./a.out 5 7 → Output: Sum = 12