Basics

- Write a program to print your name, department, and university.
- Write a program that takes two integers as input and prints their sum.
- Write a program that converts temperature from Celsius to Fahrenheit
- Write a program to calculate area and perimeter of rectangle, circle, and triangle.
- Write a program to compute simple interest and compound interest.
- Write a program to find the ASCII value of a character.
- Write a program to swap two numbers using a temporary variable
- Write a program to swap two numbers without using a temporary variable

Operators

- Write a program to check whether a number is even or odd using both modulo and bitwise operators.
- Write a program to find the maximum of three numbers using logical operators.
- Write a program to convert total days into years, weeks, and remaining days.
- Write a program to find the minimum of two numbers using the conditional (ternary) operator.
- Write a program to demonstrate the behavior of pre-increment and post-increment operators.

Decision Making

- Write a program to check whether a number is positive, negative, or neutral.
- Write a program to check whether a given year is a leap year or not.
- Write a program to find the grade of a student based on marks.
- Write a program to determine whether a character is a vowel, consonant, digit, or special symbol.
- Write a program to create a simple calculator using switch–case statements.
- Write a program to find the greatest number among three numbers using if–else statements.
- Write a program to use goto to repeatedly take input from the user until 0 is entered.

Loops

- Write a program to print numbers from 1 to 100 using all three loops (for, while, and do–while).
- Write a program to find the sum of the first N natural numbers.
- Write a program to calculate the factorial of a given number.
- Write a program to reverse a given number.
- Write a program to count the number of digits and find the sum of digits in a given number.
- Write a program to generate and display the multiplication table of any given number.
- Write a program to print all even numbers between two given limits.
- Write a program to check whether a given number is a Prime number.
- Write a program to check whether a given number is a Perfect number.
- Write a program to check whether a given number is a Strong number.
- Write a program to check whether a given number is a Spy number.
- Write a program to check whether a given number is an Armstrong number.
- Write a program to check whether a given number is a Palindrome number.

Patterns

- Write a program to print a right-angled triangle pattern of stars based on the number of rows entered by the user.
- Write a program to print a pyramid pattern of stars using nested loops.
- Write a program to print an inverted right-angled triangle pattern of stars.
- Write a program to print Floyd’s triangle up to a given number of rows.
- Write a program to print Pascal’s triangle up to a given number of rows.
- Write a program to print a hollow square pattern of stars.
- Write a program to print a diamond-shaped pattern of stars.

Arrays

- Write a program to read and print the elements of a one-dimensional array.
- Write a program to find the maximum and minimum elements in an array.
- Write a program to calculate the sum and average of the elements in an array.
- Write a program to reverse the elements of a one-dimensional array.
- Write a program to count the number of positive, negative, zero, even, and odd elements in an array.
- Write a program to perform linear search and binary search on an array.
- Write a program to sort the elements of an array using the bubble sort and selection sort algorithms.
- Write a program to add two matrices of the same order.
- Write a program to find the transpose of a matrix.
- Write a program to multiply two matrices.

Strings

- Write a program to read and print a string using gets() and puts() functions.
- Write a program to read and print a string using scanf() and printf() functions.
- Write a program to find the length of a string manually and using the strlen() function.
- Write a program to copy one string into another without using library functions.
- Write a program to concatenate two strings into one.
- Write a program to compare two strings and display whether they are equal or not.
- Write a program to count the number of vowels, consonants, digits, and spaces in a given string.
- Write a program to reverse a string manually (without using strrev()).
- Write a program to check whether a string is a palindrome.
- Write a program to convert all characters in a string to uppercase and lowercase.
- Write a program to remove all spaces from a given string.

Functions

- Write a program to create a function that calculates the sum of two numbers.
- Write a program to create a function that checks whether a given number is even or odd.
- Write a program to create a function that computes the factorial of a given number.
- Write a program to create a function that checks whether a given number is prime or not.
- Write a program to create a function that returns the grade of a student based on marks.

Recursion

- Write a program to compute power(a, b) using recursion (handle b == 0, b > 0).
- Write a program to find the factorial of a given number using recursion.
- Write a program to generate the Fibonacci series up to n terms using recursion.
- Write a program to find the Greatest Common Divisor (GCD) of two numbers using recursion.
- Write a program to compute the sum of digits of a number using recursion.
- Write a program to print a number in reverse using recursion.
- Write a program to perform recursive binary search on a sorted array.
- Write a program to solve Tower of Hanoi for n disks and print the moves.
- Write a program to print all subsets of a small set {1..n} using recursion ($n \leq 10$).

Pointers

- Write a program to display the address and value of a variable using a pointer.
- Write a program to demonstrate the difference between call by value and call by reference using pointers.
- Write a program to swap two numbers using pointers.
- Write a program to add two numbers using pointers.
- Write a program to find the length of a string using pointers.
- Write a program to reverse a string using pointers.
- Write a program to access and display the elements of an array using pointers.
- Write a program to demonstrate the concept of pointer to pointer.

Preprocessor and Macros

- Write a program to use object-like macros to define PI and compute the area of a circle.
- Write a program to create a function-like macro MAX(a,b) and demonstrate its pitfalls with expressions.
- Write a program to show conditional compilation using #ifdef DEBUG for debug prints.
- Write a program to demonstrate header guards by creating mymath.h with guards and including it twice.
- Write a program to replace an inline function with a macro SQR(x) and compare behaviors.
- Write a program to use the stringizing (#) and token-pasting (##) operators in macros.
- Write a program to print the current file, function, and line using __FILE__, __func__, and __LINE__.
- Write a program to compile alternate code paths using #if/#elif/#else based on a version macro.

Dynamic Memory Allocation

- Write a program to dynamically allocate an integer array of size n, read values, and print sum/average.
- Write a program to grow a dynamic array with realloc() as the user keeps entering numbers (end with -1).
- Write a program to allocate a 2D matrix using malloc() (array of pointers) and print its transpose.
- Write a program to read an arbitrary-length line safely by expanding a dynamic buffer.
- Write a program to remove duplicates from a dynamic array and shrink it with realloc().
- Write a program to dynamically store student records (name, roll, CGPA) and search by roll.
- Write a program to demonstrate memory leaks and then fix them using free() (before/after).
- Write a program to compare malloc() vs calloc() initialization effects.

Structures and Unions

- Write a program to create a structure for a student (name, roll, CGPA) and display the details.
- Write a program to input the details of 5 students and print the student with the highest CGPA.
- Write a program to perform addition of two complex numbers using a structure.
- Write a program to create a nested structure and display the combined information.
- Write a program to demonstrate the difference between a structure and a union..

File Handling

- Write a program to write text to a file and then read the same text from the file.
- Write a program to count the number of lines, words, and characters in a text file.
- Write a program to copy the contents of one file to another file.
- Write a program to append new data to an existing text file.
- Write a program to store and display student records using a file (add, list).

Custom Libraries

- Write a program to split code into mymath.h and mymath.c (add, sub, mul) and use them from main.c.
- Write a program to build a static library libmymath.a from mymath.c and link it in main.c.
- Write a program to mark an internal helper static in mymath.c and explain linkage.
- Write a program to declare a global in one file and access it in another using extern.
- Write a program to compile multiple C files and link them into one executable (no IDE, command line only).
- Write a program to create and use a simple header-only utility (min, max, clamp) with inline functions.
- Write a program to simulate a tiny module API (vector.h/.c) and a test main.c that uses it.

Integration Problems

- Write a program to design a menu-driven student management system using arrays, structures, and file handling.
  Features: add, search, update, delete, list, save/load from file.
- Write a program to implement a library management system using structures and file handling.
  Features: add book, issue/return, search by title/author, list.
- Write a program to simulate basic banking operations using functions and file handling.
  Features: create account, deposit, withdraw, balance inquiry, transaction log.
- Write a program to build a simple quiz game using functions, files, and strings.
  Features: load questions from file, ask MCQs, score calculation, save results.