

PROGRAMMING FOR PROBLEM SOLVING LABORATORY

Practical List – 2026–27

Subject Name	Programming for Problem Solving Laboratory
Subject Code	ICT-101P

S. No.	Practical Statement
1	Write a C program to input a student's marks in three subjects and find the total, average and percentage using suitable data types and operators.
2	Develop a C program for an online shopping bill to calculate the total cost, compare product prices, check discount eligibility and perform different types of operators.
3	Develop an algorithm and flowchart, and write a C program to check whether a given year is a leap year or not using conditional statements.
4	Write a C program to find the number of digits, sum of digits, reverse of a number and check whether the number is a palindrome using loops.
5	Develop a menu-driven C program using functions to perform addition, subtraction, multiplication and division of two numbers.
6	Write a C program to store n numbers in an array and find the largest value, smallest value, sum and average using a function.
7	Write a C program to sort a list of numbers and search for a given number using suitable sorting and searching techniques.
8	Write a C program to perform basic string operations such as finding length, copying, joining and comparing two strings.
9	Write a C program using structures to store student details and display the student having the highest total marks.
10	Write a C program to access and modify array elements using pointers and display the elements using array and pointer notation.
11	Write a C program to dynamically allocate memory for n integers, find their sum and average, and release the allocated memory.
12	Write a C program to add two matrices and store the result in a file. Read the result from the file and display it.
13	Write a C program to create a singly linked list and perform insertion, deletion and display operations.
14	Write a C program to implement a stack using an array and use it to check whether the parentheses in an expression are balanced.
15	Write a C program to implement a circular queue using an array and perform insertion, deletion and display operations.
16	Write a C program to create a binary tree and display its elements using inorder, preorder and postorder traversal.

Note: The list contains 4 practicals from each of the four units, giving a total of 16 practicals as required for ICT-101P.

BEYOND CURRICULUM – ADVANCED SKILL ENHANCEMENT PRACTICALS

The following four practicals are additional skill-enhancement exercises beyond the core 16 practicals. They are designed to develop higher-order programming skills, particularly Bloom's Taxonomy L5 (Evaluate) and L6 (Create).

S. No.	Practical Statement	Bloom's Level
17	Given a C program with syntax, logical and run-time errors, identify and correct the errors, test the program with different inputs, and improve the solution.	L5 – Evaluate
18	Develop two different C solutions for the same problem, compare their execution steps or number of operations, and justify which solution is more efficient.	L5 – Evaluate
19	Design and develop a menu-driven Student Management System using structures, functions and file handling to add, search, update and display student records.	L6 – Create
20	Choose a simple real-world problem and develop a complete C-based solution including problem definition, algorithm/pseudocode, flowchart, program, test cases and demonstration.	L6 – Create

Recommended structure: Practical 1–16 constitute the core university-aligned list (4 from each unit). Practical 17–20 are additional advanced exercises intended to strengthen debugging, testing, efficiency analysis, application development and independent problem solving.