Government College University, Lahore

**Data Structures and Algorithms**

**BSCS Semester-IV**

**LAB -1**

# Total Marks: 10 Time: 90 Minutes

**Problem 1:** Write a C++ program that does the following:

* Prompts the user to enter the size of an array (up to a reasonable limit, e.g., 100).
* Dynamically allocates an integer array of the specified size.
* Prompts the user to enter the elements of the array.
* Calculates and displays the following statistics:
  + The sum of all elements.
  + The average of all elements.
  + The largest and smallest elements.
  + The number of even and odd elements.
* Deallocates the dynamically allocated array.

**Problem 2:** A teacher has a class of students. Each student has a name and a score. The teacher wants to generate a ranked list of students based on their scores, with the highest score at the top. If two students have the same score, the student who comes first alphabetically should be ranked higher.

You are given two arrays:

* names: An array of strings representing the names of the students.
* scores: An array of integers representing the scores of the students (corresponding to the names array).

Your task is to write a C++ program that uses the Bubble Sort algorithm to sort the students based on their scores (descending) and then alphabetically by name (ascending) in case of ties. The program should output the ranked list of students, showing their name and score.

**Example Input:**

names = {"Alice", "Bob", "Charlie", "David", "Eve", "Frank"}

scores = {85, 92, 78, 92, 80, 85}

**Expected Output:**

David: 92

Bob: 92

Alice: 85

Frank: 85

Eve: 80

Charlie: 78

**Problem 3:** A university's **Examination Department** maintains a **marksheet database** where student scores are stored in a **matrix format**. Each **row** represents a student, and each **column** represents a subject. However, for administrative purposes, the department wants to **rearrange the data** such that **each row represents a subject and each column represents a student**.

your task is to write a **program that takes an input matrix (student-wise marks) and computes its transpose (subject-wise marks)**.

Enter number of students: 3

Enter number of subjects: 4

Enter marks matrix:

45 78 89 90

56 88 92 75

67 65 85 80

OUTPUT:

Original Marks Matrix (Student-wise):

45 78 89 90

56 88 92 75

67 65 85 80

Transposed Marks Matrix (Subject-wise):

45 56 67

78 88 65

89 92 85

90 75 80

.