Lab Performance Test 1

Student ID: Course Code: **CSE-244** Duration: **2 hours**

Instructions:

1. Create six source files named 2104XXX_A.cpp, 2104XXX_B.cpp, 2104XXX_C.cpp, 2104XXX_D.cpp, 2104XXX_E.cpp, 2104XXX_F.cpp. (XXX is equal to your student id). However, you are free to use any programming language and then only the file extension will be different.
2. Write the solution program of the corresponding problem in these files. We will check these files only.
3. Upload your solution program in the google classroom.
4. **Do not use internet (Browser, Whatsapp, Chatgpt, etc) during 1 Hour exam. If we find this, you will be expelled from this test.**

Problem A: Occurrences of a Number

Description:

Given an unsorted list of numbers, your task is to determine how many times a specific number occurs in the list. The solution must have a time complexity of O(n).

Input Format:

- The first line contains an integer n, the number of elements in the list.
- The second line contains n integers, representing the elements of the unsorted list.
- The third line contains a single integer xxx, the specific number to find in the list.

Output Format:

A single line that states how many times x appears.

Sample Input Output:

Input	Output
6 -4 9 4 9 9 31 9	3 times
5 1 2 3 4 5 6	0 times

Problem B: Subarray Sum

Description:

Given an array of non-negative integers, and a value sum, determine if there is a subset of the given set with sum equal to given sum.

Input Format:

The program will read an integer variable ***N***, followed by ***N*** integer numbers (**arr**) and finally an integer variable ***sum***.

Constraints:

$0 < N <= 100$, $0 <= arr[i] <= 1000$, $0 < sum <= 100000$

Sample Input Output:

Sample Input	Sample Output
6 3, 34, 4, 12, 5, 2 9	1
6 3, 34, 4, 12, 5, 2 30	0

Problem C: Cutting a Rod Problem

Description:

Given a rod of length n inches and an array price[]. price[i] denotes the value of a piece of length i. The task is to determine the maximum value obtainable by cutting up the rod and selling the pieces.

Input Format:

The program will read one integer variable ***n,*** where ***n*** represents the length of the rod. Then, there will be n different integer number denote the value of a piece of length i.

Constraints:

0 < ***n, price[i]*** <= 1000

Sample Input Output:

Sample Input	Sample Output
8 1, 5, 8, 9, 10, 17, 17, 20	22
8 3, 5, 8, 9, 10, 17, 17, 20	24
1 3	3

Problem D: Edit Distance

Description:

Given two strings **s1 and s2** of lengths m and n respectively and below operations that can be performed on s1. Find the minimum number of edits (operations) to convert **'s1' into 's2'**.

Insert: Insert any character before or after any index of **s1**

Remove: Remove a character of **s1**

Replace: Replace a character at any index of **s1** with some other character.

Input Format:

The program will read two string **s1 and s2.**

Constraints: The length of the two string must be less than 100 characters.

Sample Input Output:

Geek Gesek	1
Cat Cut	1
Sunday Saturday	3

Problem E: Detecting Cycle

Description: You are given an undirected graph with **n** nodes and **m** edges. Your task is to implement a program to check if there is a cycle in the graph using Depth-First Search (DFS). If a cycle is found, output YES; otherwise, output NO cycle.

Input Format:

- The first line contains two integers, n and m, representing the number of nodes and edges in the graph.
- The next m lines contain two integers each, a and b, denoting an undirected edge between nodes a and b.

Output Format:

- Output YES if there is at least one cycle in the graph.
- Output NO cycle if the graph is acyclic.

Constraints:

The graph can contain multiple components.

Sample Input Output:

Input	Output
5 5 1 2 2 3 3 4 4 5 5 1	YES
4 2 1 2 3 4	NO cycle

Problem F: Minimum Spanning Tree (MST)

Description:

You are given an undirected, weighted graph with **n** vertices and **m** edges. Your task is to calculate the weight of the Minimum Spanning Tree (MST) of the graph.
A Minimum Spanning Tree is a subset of the edges in a graph that connects all vertices together without any cycles and with the minimum possible total edge weight.

Input Format:

Two integers, n and m, where:

- n is the number of vertices ($1 \leq n \leq 10{,}000$).
- m is the number of edges ($1 \leq m \leq 100{,}000$).

The next m lines contain three integers a, b, and w:

- a and b are the endpoints of an edge ($1 \leq a, b \leq n$).
- w is the weight of the edge ($1 \leq w \leq 1{,}000{,}000$).

Output Format: A single integer, the weight of the Minimum Spanning Tree (MST).

Constraints: The graph may have multiple edges and weights, but no self-loops.

Sample Input Output:

Input	Output
4 5 1 2 7 1 4 6 4 2 9 4 3 8 2 3 6	19
3 3 1 2 10 2 3 15 1 3 5	15