INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CSC-101 PROGRAMMING WITH C AND C++

Autumn 2023 Problem Set 05

1. You are given an integer array ‘arr’ of size ‘n’ where ‘arr[i]’ represents the height of the ‘i-th’ building in a city. Due to the shadows created by buildings, you are interested in finding out how many buildings have a view of the sunset. A building has a view of the sunset if there are no buildings taller than it to its right.

   Write a C program that:

   1. Reads the number of buildings, ‘n’, from the user.

   2. Reads the heights of these buildings from the user.

   3. Calculates and prints the number of buildings with a view of the sunset.

   Example:

   Input:

   6

   3 7 8 3 6 1

   Output:

   3

   Explanation: The buildings with heights 8, 6, and 1 have a view of the sunset.

2. The city is built on a two-dimensional plane. Streets are horizontal and vertical lines that divide the city into blocks. At certain intersections, there are water fountains. Given the coordinates of the starting point, ending point, and the fountains, determine the minimum distance you have to walk to reach the destination while passing by at least one fountain. For simplicity, consider walking only along streets (horizontal or vertical lines), not diagonally.

Write a C++ program that:

1. Reads the number of fountains, ‘n’.

2. Reads the coordinates of the starting point ‘startX, startY’.

3. Reads the coordinates of the ending point ‘endX, endY’.

4. Reads the ‘x’ and ‘y’ coordinates of each of the ‘n’ fountains.

5. Calculates and prints the minimum distance to walk to the ending point, ensuring you pass by at least one fountain.

Input Constraints:

- $1 \leq \text{n} \leq 10^4$.
- $-10^5 \leq$ startX, startY, endX, endY, fountainX, fountainY $\leq 10^5$.

3. In a board game, each player’s score is represented as a string of lowercase English alphabets, where each character represents a unique scoring criteria (e.g., ‘a’ might represent capturing a castle, ‘b’ might represent finding a treasure, and so on). The value of each character is its alphabetical position (‘a=1, b=2, . . . , z=26’). Players wish to boost their scores by using power-ups. Each power-up allows a player to duplicate one character in their score string. Given a player’s score string and a number of available power-ups, write a C++ program to determine the maximum possible score the player can achieve. The input to the program will be a string and an integer, and the output should be the maximal score.

   Example:

   Input:

   score string: “abc”

   power-ups: 2

   Output:

   11

Explanation: Use one power-up to duplicate ‘c’ (3 points) and another to duplicate ‘b’ (2 points). The new string will be “abccb”, and the score will be 1 + 2 + 3 + 3 + 2 = 11.

4. In a kingdom, each knight is assigned a unique code, represented by a string of even length made up of pairs of lowercase English alphabets. Each pair of characters represents a specific attribute of the knight (e.g., ‘ab’ might mean the knight is brave, ‘cd’ might mean he’s clever, etc.). Two knights can form a duo if they share exactly one common attribute and are different in all others.

   Given a list of knight codes, write a C++ program to determine how many unique duos can be formed. The input to the program will be a vector of strings, and the output should be an integer indicating the number of unique duos.

   Example:

   Input:

   knight-codes: “ab”, “cd”, “ad”, “bc”

   Output:

   4

   Explanation: Duos that can be formed are:

   1. “ab” and “ad”

   2. “ab” and “bc”

   3. “cd” and “ad”

   4. “cd” and “bc”

5. In a virtual world, every person has an aura represented by a string of lowercase English letters. An aura match between two people is determined by the following rules:

   1. The aura of both individuals must be of the same length.

   2. Both auras are a match if the number of distinct characters in one aura is equal to the number of continuous repeating sequences in the other aura.

For example, the aura “aaabcc” matches with “efefef” because:

- Both auras have 6 characters.
- “aaabcc” has 3 distinct characters (‘a’, ‘b’, ‘c’) and “efefef” has 3 continuous repeating sequences (“ef”, “ef”, “ef”).

Write a C++ program that reads two auras and determines if they are a match based on the rules above. The program should output “Match” if the auras match and “No Match” otherwise.

6. Imagine a realm where numbers converse in a unique dialect. In this dialect, every number expresses itself as a palindrome. However, some numbers are forbidden to speak in the conventional sense. They communicate using palindromes that can be formed by reversing their half-length. For instance, the number ‘1234’ communicates as ‘1221’, but the number ‘12345’ communicates as ‘12321’.

   Your task is to write a C++ program that translates given integers into this peculiar dialect. The program should take an integer as input and return its representation in the palindrome dialect. If an integer is already a palindrome, it should be left unchanged.

7. In the land of Synthetica, people communicate using synthesized sounds. Each sound is represented by a unique integer frequency. Synthetica has an age-old tradition: During a conversation, if two people use the same sound frequency consecutively, they must introduce a silence (denoted by ‘0’) before continuing the conversation.

   For instance, if the sequence of frequencies in a conversation is ‘1, 2, 3, 3, 4’, it should actually be spoken as ‘1, 2, 3, 0, 3, 4’.

   Write a C++ program that takes an input sequence of frequencies as a list of integers. The program should insert ‘0’ wherever needed based on the tradition of Synthetica and output the modified sequence.

8. In a fantasy video game, each player can hold a maximum of two weapons at a time. The efficiency of a weapon combo is the sum of their individual efficiencies minus the product of their overlap percentage. Given a list of weapons with their efficiencies and overlap percentages with every other weapon, determine the weapon combo that yields the highest efficiency. Write a C++ program that reads the number of weapons, their efficiencies, and their overlap percentages with each other, then calculates and prints the two weapons forming the optimal combo.

   Input Constraints:

   - $2 \leq$ number of weapons ≤ 50.
   - Efficiency values are positive integers up to 1000.
   - Overlap percentage is a float between 0 and 1, inclusive.

   Note: The overlap percentage is symmetric (i.e., overlap between weapon A and weapon B is the same as overlap between weapon B and weapon A).

9. You are given an 'n × n' square matrix filled with integers. Your task is to write a C program that reads this matrix and then prints the pattern formed by the border elements of the matrix in a spiral order, starting from the top-left corner and moving in a clockwise direction.

   Input Constraints:

   - The matrix size 'n': $1 \leq$ n ≤ 100.
   - Matrix elements: $-10^4 \leq$ matrix[i][j] $\leq 10^4$.

   Example: Input:

   ```
   3
   1 2 3
   4 5 6
   7 8 9
   ```

   Output: 1 2 3 6 9 8 7 4

10. In a land of magic, there exists a peculiar type of matrix known as a Harmonic Matrix. A Harmonic Matrix is an ‘n × n’ square matrix such that:

    1. The first row is filled with the number 1.

    2. The second row is filled with the number 2.

    3. The third row is filled with the number 3.

    . . . and so on, until the nth row.

    However, there’s a twist. Starting from the 2nd row, for each column, if the previous column’s value in that row is divisible by the row number, then the next column’s value for that row will be the previous column’s value + 1. If not, it will wrap around to 1.

    For example, in a ‘3 × 3’ Harmonic Matrix:

    ```
    1 1 1
    2 3 1
    3 4 1
    ```

    Your task is to write a C program that generates the Harmonic Matrix for a given ‘n’.

    Input Constraints: The matrix size ‘n’: 1 ≤ n ≤ 50.

    Example:

    Input: 3

    Output:

    ```
    1 1 1
    2 3 1
    3 4 1
    ```