INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CSC-101 PROGRAMMING WITH C AND C++

Assignment 05 Autumn 2023

1. You are provided with a square matrix (2D array) of positive integers, with dimensions 'N x N'. Your mission is to design an algorithm that will shift all non-zero elements of the matrix to the left side of each row, maintaining the relative order of the numbers, while ensuring that all zeros in a row are positioned to the right. Your transformed matrix should still be 'N x N', and the sequence of the non-zero numbers in each row must remain the same. After the transformation, write a function to calculate the sum of the primary diagonal (from the top-left to bottom-right) of the transformed matrix. Provide the transformed matrix and the sum of its primary diagonal as output. For instance, if given a matrix '[[4, 0, 2], [0, 3, 0], [1, 0, 0]]', your program should return the transformed matrix '[[4, 2, 0], [3, 0, 0], [1, 0, 0]]' and a primary diagonal sum of '4'. How will you achieve this, and can you optimize it for speed?

2. Imagine a world where strings of text are used as melodies in music. A string's melody is determined by the sequence of its characters. You're tasked with writing a program that identifies "harmonious sequences" within a given string. A sequence is harmonious if it contains at least three consecutive characters that are also consecutive in the alphabet, irrespective of their case. For instance, the string "ABc" or "ghI" would be harmonious, while "acb" wouldn't be. Your program should be able to process a given text and extract all harmonious sequences it contains. For an input like "MusABCal note and the DEFGh", your program should identify and display the sequences like "ABc" and "DEFGh". What approach would you take, and how would you ensure that overlapping sequences are not missed?

3. In a virtual simulation, you find yourself in a mysterious garden. The garden is represented as a grid of 'M x N' cells. Each cell contains a flower with a unique fragrance intensity value, which is always a positive integer. Your task is to create a program that determines the path through the garden which maximizes the total fragrance collected. The constraints are as follows:

   - You start from the top-left corner and finish at the bottom-right.

   - At each step, you can only move right or down.

- As you walk, you collect the fragrance of every flower you step on.

Write a program that can determine the maximum fragrance intensity you can collect from the top-left to the bottom-right. Can you also determine the exact path you took?

For example, given a garden grid: '[[1, 3, 2], [4, 8, 1], [1, 5, 3]]', the optimal path is '[1, 4, 8, 5, 3]' with a total fragrance intensity of '21'.

4. You've been given an array of positive integers. Your task is to create a program that identifies the largest contiguous sub-array that forms a palindrome. If there are multiple sub-arrays with the same length, choose the one that appears first. Additionally, your program should calculate the sum of the elements in this sub-array. For instance, if your input array is '[1, 2, 3, 8, 7, 8, 3, 2, 1, 4, 5]', the largest contiguous palindromic sub-array is '[1, 2, 3, 8, 7, 8, 3, 2, 1]' and its sum is '35'.

5. Given two arrays, 'A' and 'B', each of size 'N', your task is to generate an array 'C' such that each element of 'C' at index 'i' is the sum of the 'i-th' Fibonacci number multiplied by the 'i-th' element of 'A' and the '(i+1)-th' Fibonacci number multiplied by the 'i-th' element of 'B'. In mathematical terms, 'C[i] = F[i] * A[i] + F[i+1] * B[i]' where 'F[i]' represents the 'i-th' Fibonacci number. However, there's a catch! You must achieve this without directly computing the Fibonacci sequence in a separate array or list, and you need to optimize for space. For example, if 'A = [2, 3, 4]' and 'B = [5, 6, 7]', the resulting 'C' should be '[7, 15, 29]'. Can you come up with an efficient approach to tackle this problem and generate the 'C' array?

6. Imagine being handed a lowercase string, 's', and your challenge is to split this string into a set of substrings, ensuring that each substring is a palindrome. The twist is that you need to minimize the number of cuts, or partitions, you make. Create a program that determines the minimum number of cuts needed and displays a possible partitioning solution. For example, for the string "racecarbb", a possible optimal solution is to split the string into ["racecar", "b", "b"], requiring only 2 cuts.

7. You are an adventurer standing at the top-left corner '(0,0)' of a 2D grid of size 'M x N', where each cell either contains a positive integer (the amount of treasure) or a '0' (indicating it's empty). Your goal is to collect as much treasure as possible and reach the bottom-right corner '(M-1, N-1)' of the grid. However, there's a catch!

At any given cell, you can only move right or down. Additionally, if you choose to take the treasure in a cell, the total treasure in its immediate right and immediate down cells gets doubled. If you skip the treasure, the values in the neighboring cells remain unchanged. Your task is to devise an algorithm to determine the path that will maximize your treasure collection. Describe the route you'll take (e.g., "right, down, down, right") and the total amount of treasure collected. Remember, you can skip treasures if it helps increase your total haul. Can you strategize the optimal path to amass the maximum bounty?

8. Imagine being handed a list of numbers with a unique pattern. Your objective is to design a program that detects the largest continuous increasing subsequence from this list and then calculates the product of its elements. The catch? Some numbers in the list might be missing, represented by a '-1'. For these missing numbers, you may replace them with any positive integer of your choice to optimize for the longest increasing subsequence. Once you determine this optimal sequence, compute its product. For example, if given the list '[3, 4, -1, 6, 7]', you might replace '-1' with '5' to have an increasing subsequence of '[3, 4, 5, 6, 7]' and a product of '2520'.

9. Imagine you're working on a graphical software where every color is represented as an RGB triplet. For the sake of this problem, each value in the RGB triplet is a positive integer between 0 and 255. Now, you are presented with a unique challenge. You need to develop an algorithm that takes in two colors as input, say C_1 and C_2, and generates a sequence of 'N' colors that transition from C_1 to C_2. The transition between the two colors should be smooth, and each subsequent color in the sequence should be derived by evenly incrementing (or decrementing) the RGB values from C_1 to C_2. The first color of the sequence should be C_1 and the last one should be C_2. Your task is not just to produce this sequence, but also to find a color, C_x, in your generated sequence such that the sum of its RGB components is the highest. Create the sequence and pinpoint C_x. For a given N, can you ensure your solution is efficient?

   Note: In the context of computer graphics and color theory, a "smooth transition" between two colors means that the change from one color to the other occurs in a gradual manner, without abrupt or sudden shifts. This is often visualized as a gradient where the color flows seamlessly from the starting color to the ending color. For RGB colors, a smooth transition implies that the Red, Green, and Blue components of the two colors change at a consistent rate. In practical terms, if you're transitioning from color C_1 with RGB values (R_1, G_1, B_1) to color C_2 with RGB

values (R_2, G_2, B_2) over 'N' steps, then the change in each of the RGB components for each step would be:

$$\Delta R = \frac{R_2 - R_1}{N - 1}$$
$$\Delta G = \frac{G_2 - G_1}{N - 1}$$
$$\Delta B = \frac{B_2 - B_1}{N - 1}$$

Each step in the sequence would increment (or decrement) the RGB values of C_1 by ΔR, ΔG, and ΔB respectively, until C_2 is reached. This results in a set of colors that perceptually move from the starting color to the ending color without any jarring changes, providing the viewer with a pleasing and continuous visual experience.

10. Imagine you're an archaeologist working on deciphering ancient manuscripts. One peculiar script you encounter is written in lines, but there's a twist. Each line is a mix of letters and numbers, and the meaning of the line can only be understood when read in reverse order. However, in this ancient language, the position of numbers within each line is significant and must remain unchanged while the letters alone are to be reversed. Your challenge is to write a program that takes in a string representing one such line from the manuscript and returns the deciphered line. For instance, given the string "a1bc23de", the output should be "e1dc23ba". Note: The input string consists of lowercase letters and numbers, and its length will not exceed 1000 characters.

Submission Link