

National University of Computer and Emerging Sciences

Data Structures (CL2001)

Date: October 16th 2024

Course Instructor(s)

Mr. Shafique Rehman

Ms. Zainab Asif Jawed

Mr. Misbah

Lab Mid Exam(B)

Total Time: 120 minutes

Total Marks: 50

Total Questions: 03

Semester: Fall-2024

Campus: Karachi

Dept.: Computer science

- You must comment your student ID on top of each file in the .cpp and word file.
- Name the .cpp file for each question according to Roll_No e.g. k23-xxxx_Q1.cpp, k23-xxxx_Q2.cpp etc.
- Create a ZIP folder of all your solutions and copy it in the local storage with the title K23-xxxx_B.
- Submission are on local storage that can be accessed using win+r keys and entering \\172.16.5.41
- address in the dialog box.
- Enter your username as khifast\K23xxxx and its assigned password.
- Zip folder needs to be pasted in the "Exam Submission\teacherName\CourseName.

Student Name

Roll No

BCS-30

Section

Student Signature

LLO # 1: Use & explain concepts related to basic and advanced data structures and describe their usage in terms of common algorithmic operations. Marks: [8 wtg. 16 marks]

Given two arrays, your task is to sort the first array according to the frequency of its elements as specified by the second array. The elements of the first array that are present in the second array should appear first, sorted by their frequency in descending order. If two elements have the same frequency, they should appear in the order they were first encountered in the first array. Elements not present in the second array should appear afterward in their original order.

For Example: First array: {5, 3, 5, 2, 3, 4, 2}, Second array: {3, 5, 2}. The Output should be {5, 5, 3, 3, 2, 2, 4} because, 5 appears twice, 3 appears twice, and 2 appears twice as well, and they are available in the second array as well. So, they are listed first based on their frequency. The number 4 is listed last because it is not in the second array. See Example 2 for better understanding

First array: {5, 3, 5, 2, 3, 4, 2}

Second array: {5, 2}

Output: {5, 5, 2, 2, 3, 3, 4}

In the first array 5 appears twice. 2 appears twice. Since both 5 and 2 appear the same number of times, we keep their original order from the first array. The remaining elements in the first array are 3 and 4. These elements are not in the second array and should maintain their original order.

National University of Computer and Emerging Sciences

LLO # 2: Compare different data structures in terms of their relative efficiency and design effective solutions and algorithms that make use of them. Marks: 18 wtg, 16 marks]

A deck of cards has 52 playing cards. Each card has 2 attributes: rank & suit. The rank can be represented by a number (2-14). 2-10 as normal and 11,12,13,14 will represent J, Q, K, A respectively. The suit can be represented by a character. 'h' for hearts, 'd' for diamonds, 'c' for clubs and 's' for spades. Create a class **Card** to represent this.

Assume that the **randomly generated** cards will be unique (no need to check). To generate the cards randomly, simply generate **random** values based on which **rank & suit** will be assigned.

You will simulate a game of cards between 2 players, where each player's cards will be stored within a separate linked list. Each node in the linked list will contain a **Card** object. Players take turns placing cards from their hands. until 7 rounds are played.

After all, two players place a card, the cards are compared:

- If all two cards match in **suit**: The player with the lowest rank card will place the next card from their hand. The player with highest rank will get 2 points and last player will not get any point.
- If the cards have different **suits**: The player with the highest **suit** rank card will collect all three cards, adding them back to their hand. And his points will be decreased. **Suits** are ranked in the following order: Clubs, Spades, Hearts, Diamonds. Meaning ALL Clubs are more valuable than any Spades, and so on.
- If any player plays any an Ace ('A' or rank 14), they automatically get 5 points, regardless of the other cards played. The game continues in rounds until 7 rounds are played.

After 7 rounds, the player who has more cards than the other player will lose 3 points.
The player with the most points overall will be declared the winner.

LLO # 3: Use & explain concepts related to basic and advanced data structures and describe their usage in terms of common algorithmic operations. Marks: 9 wtg, 18 marks]

A treasure hunter is exploring an ancient cave and needs to find their way to the hidden treasure room. They are currently located at the top left corner of a 6x6 matrix. The matrix represents the area, with 'D' indicating dangerous areas and 'H' indicating hazardous zones. The treasure hunter starts with full energy and will lose some energy when entering a hazardous zone. They cannot enter a dangerous area (marked with 'D').

Find a path and calculate the energy for the treasure hunter to reach the treasure room (T) while maximizing their energy and avoiding dangerous areas (D).

S	P	P	D	H	H
P	P	P	D	P	H
P	H	P	P	P	D
P	D	P	D	P	P
P	P	P	H	P	D
P	D	P	P	P	T

Note: 'S' marks the starting point, 'T' is the treasure room, 'D' signifies dangerous areas, 'H' represents hazardous zones, and 'P' denotes safe paths.