|  |  |
| --- | --- |
| **logo** | **COMSATS University Islamabad**  **Department of Computer Science**  **Assignment 1 [Marks 10]** |

Class: BSAI-3 Submission Date: 30-09-2024

Subject: (CSC211) Data Structures Instructor: Dr. Zobia Rehman

**Theory Assignment-1:**

**CLO-1: SO(1,2):Employ linear data structures to solve computing problems.**

What to do:

a) Write the steps to perform the task.

b) Diagrammatically show the impact of each important step.

c) Write the code against that step.

**Lab Assignment-1:**

**CLO-4: SO(2,3,4): Implement linear & non-linear data structures, sorting & searching algorithms and hashing.**

What to do:

Implement the given tasks in C/C++ language.

**Tasks of Singly Linked List:**

1. Print the singly linked list in reverse order by iteration (without using any additional array/List/linked list etc.).
2. Print the singly linked list in reverse order by recursion (without using any additional array/List/linked list etc.).
3. Reverse the linked list (before and after the operation, address of each node will remain same).
4. Remove duplicates from a linked list.
5. Detect loop or cycle in a linked list.
6. Swap any two nodes (only links will change, not to swap the values, so after swap addresses of nodes will remain same).
7. Assume that you have a linked list having even and odd values. Your task is to spit the list into two separate lists of even and odd without changing their addresses. Also there should be a single method to print all linked list means you just have to pass the address of first node and it should print the list accordingly.
8. Swap any two nodes with their addresses (after swap there will be no change of address).
9. Reverse the first half and second of the linked list i.e. assume that the list is:

1->2->3->4->5->6->7->8

After calling the reverse method the list should be:

4->3->2->1->8->7->6->5

Given a singly linked list, write a function to swap elements pairwise i.e. assume that the list is:

Input : 1->2->3->4->5->6->NULL

Output : 2->1->4->3->6->5->NULL

**Task of Circular Linked List:**

1. Write the program to solve the Josephus problem. In Josephus problem you are given the total number of persons N and the number M, which indicates that M-1 persons are skipped and the Mth person is killed in the circle, your task is to choose the place in the initial circle so that you are the last one remaining (survive).

**Tasks of Doubly Linked List:**

1. Swap every alternate node of the doubly linked list starting from start and end i.e. assume that the list:

Input: 1->2->3->4->5->6->7->8->9

After calling the method the list should be:

Input: 1->8->3->6->5->4->7->2->9

1. Suppose you are working on a project to develop an inventory management system for a retail chain that has multiple stores in different cities across the country. The system should be able to maintain the records of the items in each store. Each store has different sections, and each section has different products. To implement this system, use best suitable Data Structure to maintain the records of items, locations, stores, and sections.

Write the methods to perform following operations:

➢ Add a new section in a store (toys, grocery, fruits….)

➢ Store an item in a particular section of a particular store.

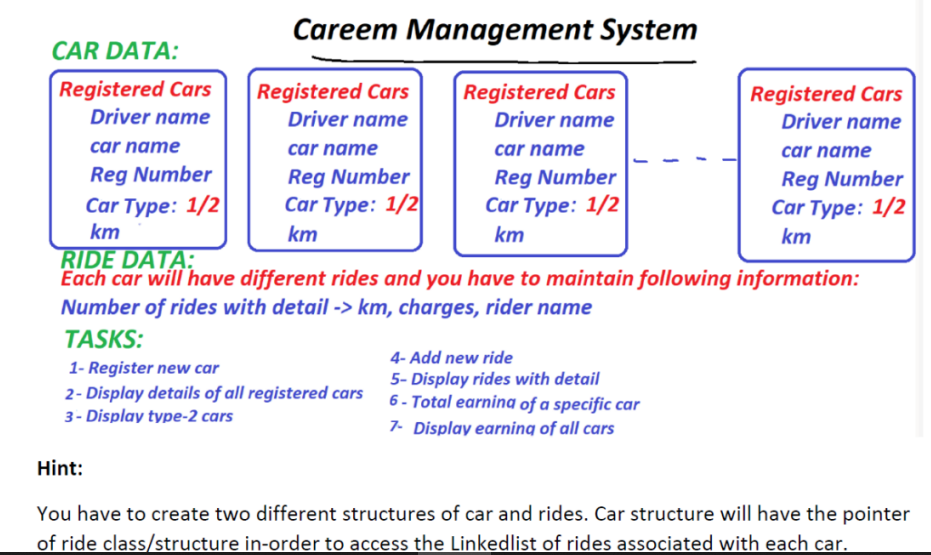
➢ Remove an item in a particular section of a particular store.

➢ Display the list of all items of a particular section of a store.

➢ Display the list of items for a given store.

For example, to store an item in a location in a particular store, you can search for the node that represents the store, then search for the node that represents the section in the store, and then search for the node that represents the location in the section. Then, you can add a new node to the linked list that represents the items stored in the location. Same procedure will be for deletion operation.

1. Implement Careem Management System explained below.



1. You are working for a company that manages flight data for multiple airports. The company stores information about flights between different airports, but only a small percentage of the potential routes between these airports have actual flights, making the data sparse. You need to implement a system to efficiently store, manipulate, and process this sparse flight data using a sparse matrix represented as a linked list. Each entry in the sparse matrix will store the number of flights between two airports (represented by rows and columns of the matrix). The system should store only non-zero values (i.e., existing flights), as the majority of routes have no flights. Implement the following operations:
2. Store flight data: Input and store only non-zero entries in the matrix using a linked list.
3. Transpose the flight matrix: Implement a function to transpose the matrix, where rows represent departure airports and columns represent arrival airports. The transpose flips the matrix, showing flights arriving at each airport instead of departing.
4. Add flight data from two matrices: Combine the flight data for two different weeks., Adding the flight data for two different weeks gives the total number of flights across both weeks.
5. Multiply flight matrices: Multiply two flight matrices to analyze connecting flights between airports via intermediate airports. This can help analyze indirect flight connections, i.e., flights from Airport A to Airport C via Airport B. This could be useful for determining connecting flights between airports.
6. Finally, write a function to print the matrix in a readable format, showing all airports and the number of flights between them.

**Explanation:**

