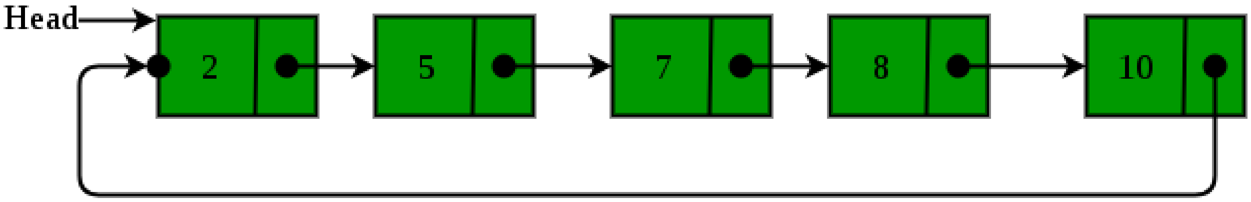
Data Structures Lab 4(A)

Course: Data Structures (CL2001) **Semester:** Fall 2023
Instructor: Sameer Faisal **T.A:** N/A

Note:

- Maintain discipline during the lab.
- Listen and follow the instructions as they are given.
- Just raise hand if you have any problem.
- Completing all tasks of each lab is compulsory.
- Get your lab checked at the end of the session.

Circular Link List



*The **circular linked list** is a linked list where all nodes are connected to form a circle. In a circular linked list, the first node and the last node are connected to each other which forms a circle. There is no NULL at the end.*

```
class Node {
        public:
                int key;
                int data;
                Node * next;

                Node() {
                        key = 0;
                        data = 0;
                        next = NULL;
                }

                Node(int k, int d) {
                        key = k;
                        data = d;
                }
};

class CircularLinkedList {
        public:
                Node * head;

                CircularLinkedList() {
```

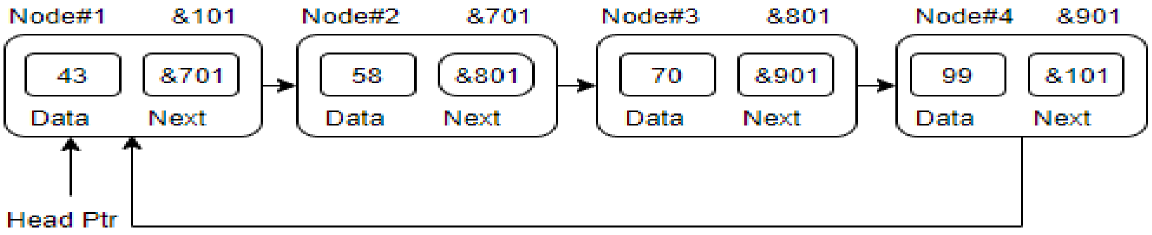
```
                head = NULL;
            }


        appendNode();
        prependNode();
        insertNodeAfter();
        deleteNodeByKey();
        updateNodeByKey();
        print();
};
```

Task-1:

Create a circular link list and perform the mentioned tasks.

i. Insert a new node at the end of the list.
ii. Insert a new node at the beginning of list.
iii. Insert a new node at given position.
iv. Delete any node.
v. Print the complete circular link list.



Task-2:

Given a linked list, you have to perform the following task:

1. Extract the alternative nodes starting from second node.
2. Reverse the extracted list.
3. Append the extracted list at the end of the original list.

Note: Try to solve the problem without using any extra memory.

Example 1:

Input:

LinkedList = 10->4->9->1->3->5->9->4

Output:

10 9 3 9 4 5 1 4

Explanation:

Alternative nodes in the given linked list are 4,1,5,4. Reversing the alternative nodes from the given list, and then appending them to the end of the list results in a list 10->9->3->9->4->5->1->4.