

UCS301 Data Structures
Lab Assignment 6
Doubly and Circular Linked List
(Week 6)

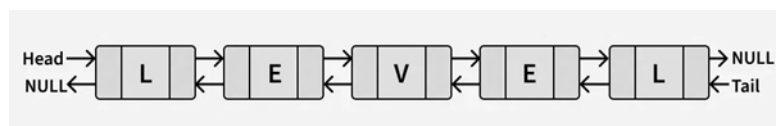
1. Develop a menu driven program for the following operations of on a Circular as well as a Doubly Linked List.
 - (a) Insertion anywhere in the linked list (As a first node, as a last node, and after/before a specific node).
 - (b) Deletion of a specific node, say 'Delete Node 60'. That mean the node to be deleted may appear as a head node, last node or a node in between.
 - (c) Search for a node.
2. Display all the node values in a circular linked list, repeating value of head node at the end too.

Input: 20 → 100 → 40 → 80 → 60,

Output: 20 100 40 80 60 20.
3. Write a program to find size of
 - i. Doubly Linked List.
 - ii. Circular Linked List.
4. Write a program to check if a doubly linked list of characters is palindrome or not.

Example 1:

Input:



Output:

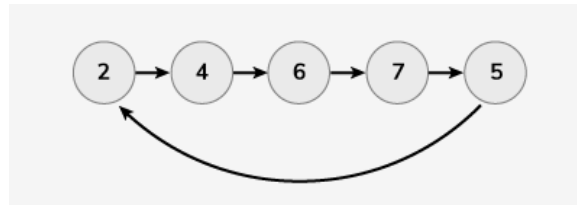
True

5. Write a program to check if a linked list is Circular Linked List or not.

Example 1:

LinkedList: 2->4->6->7->5

Input:



Output:

True

Additional Questions: Asked multiple times in FAANG+ companies

1. Given a Circular linked list. The task is split into two Circular Linked lists. If there are an odd number of nodes in the given circular linked list then out of the resulting two halved lists, the first list should have one node more than the second list.

<https://www.geeksforgeeks.org/split-a-circular-linked-list-into-two-halves/>

Input: 10->4->9

Output: 10->4 , 9

2. Given a Doubly linked list and Circular singly linked list containing N nodes, the task is to remove all the nodes from each list which contains elements whose parity is even.

<https://www.geeksforgeeks.org/remove-all-even-parity-nodes-from-a-doubly-and-circular-singly-linked-list/>

Note: Parity means no of 1 in binary form

11 -> 1011, parity = 3

15 -> 1111, parity = 4

Example 1:

Input: CLL: 9 -> 11 -> 34 -> 6 -> 13 -> 21

Output: 11 -> 13 -> 21

Example 2:

Input: DLL: 18 <=> 15 <=> 8 <=> 9 <=> 14

Output: 8 <=> 14

3. Given a Doubly linked list containing n nodes. The task is to reverse every group of k nodes in the list. If the number of nodes is not a multiple of k then left-out nodes, in the end should be considered as a group and must be reversed.

<https://www.geeksforgeeks.org/reverse-doubly-linked-list-groups-given-size/>

Example 1:

Input: 1 <-> 2 <-> 3 <-> 4 <-> 5 <-> 6 <-> NULL, k = 2

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

Example 2:

Input: 1 <-> 2 <-> 3 <-> 4 <-> 5 <-> 6 <-> NULL, k = 4

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

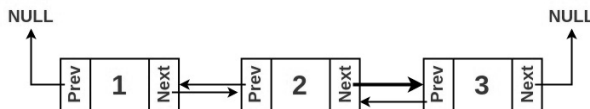
- Given a doubly linked list having exactly one of the node pointing to a random node in the list, the task is to correct this random pointer in the doubly linked list, such that it points to the expected node.

<https://www.geeksforgeeks.org/correct-the-random-pointer-in-doubly-linked-list/>

Input:



Output:



- Given a 2D matrix, the task is to convert it into a doubly-linked list with four pointers that are next, previous, up, and down, each node of this list should be connected to its next, previous, up, and down nodes.

<https://www.geeksforgeeks.org/construct-a-doubly-linked-linked-list-from-2d-matrix/?ref=rp>

Input: 2D matrix

1 2 3
4 5 6
7 8 9

Output:

