

## Lab Assignment 2: Linked List

**Instructions:** All the values has to be taken from user. For example, size of array or linked list, data of the linked list, where the node to be inserted in the linked list, etc.

For singly and doubly linked list follow the same strcuture of the node as discussed in the class as follows:

```
Struct node { int data; struct node *link;}  
strcut node { int data; struct node *fwd, *back;}
```

### Singly Linked List

1. Write a pogram (WAP) to create the linked list of n nodes.
2. WAP to insert a node in the linked list created in (1):
  - a. at the beginning b. at the end c. anywhere between first and last node given by the user.
3. WAP to delete a node in the linked list created in (1):
  - a. at the beginning b. at the end c. anywhere between first and last node given by the user.
4. WAP to reverre the linked list created in (1).
5. WAP to find the nth node from the last node of the linked list.
6. Given two sorted linked lists of n elements each, WAP for the printing common elements of them.
7. Consider two linked list both of size n nodes. Both the list contain integers in ascending order. WAP to merge these two lists so that final list should also be in ascending order. The final list should contain all distinct element.
8. Given a linked list of n noded with even and odd numbers, WAP for making changes to the list in such a way that all even numbers appear at the beginning.
9. We are given a pointer to a node (not the last node) in a linked list. Delete that node from the linked list.
10. Reverse the linked list in pairs. If you have a linked list that holds  $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow X$ , then after the function has been called the linked list would hold  $2 \rightarrow 1 \rightarrow 4 \rightarrow 3 \rightarrow X$ .

### **Doubly Linked List**

11. Write a program (WAP) to create the linked list of  $n$  nodes.
12. WAP to insert a node in the linked list created in (11):
  - a. at the beginning
  - b. at the end
  - c. anywhere between first and last node given by the user.
13. WAP to delete a node in the linked list created in (11):
  - a. at the beginning
  - b. at the end
  - c. anywhere between first and last node given by the user.

### **Circular Singly Linked List**

14. WAP to create circular linked list of  $n$  nodes.
15. WAP to count the number of nodes in circular linked list if only start pointer of circular linked list is given.
16. WAP to reverse the linked list created in (14).
17. WAP to insert a node in the linked list created in (14):
  - a. Beginning
  - b. At the end
18. WAP to delete a node in the linked list created in (14):
  - a. Beginning
  - b. At the end
19. In case of circular linked list last node is linked with first node. But, consider the case when the last node is linked with any of the remaining nodes of the linked list. So, your program should follow things:
  - a. Create such linked list of  $n$  nodes where it should be asked from the user that to which node the last node should be linked.
  - b. WAP to find the node to which the last node is linked. Consider also the case when the last node is pointing to NULL i.e. Singly linked list.
20. Split a Circular Linked List of  $n$  nodes into two equal parts. Both splitted linked lists should also be circular. If the number of nodes in the list are odd then make first list one node extra than second list.

## **Circular Doubly Linked List**

21. Write a program for circular doubly link list to perform all the operations .i.e. insertion, deletion