DSA LAB Programming Practice
Linked List

1. Write a menu driven program for inserting, deleting, and display operation on the following:
   - single linked list
   - doubly linked list
   - single circular linked list
   - double circular linked list
2. Write a non-recursive function to display the single linked list in the reverse order.
3. Write a function to reverse the nodes of a double linked list.
4. Write a program to swap every two consecutive nodes in a linked list without swapping data.
5. WAP to find nth node from the end of a Linked List.
6. Write a function to delete the whole Linked List.
7. Given a singly linked list, determine if its a palindrome. Return 1 or 0 denoting if its a palindrome or not, respectively. (without using recession)

   List 1-->2-->1 is a palindrome.

   List 1-->2-->3 is not a palindrome.

8. Given a sorted linked list, delete all duplicates such that each element appear only once.

   For example, Given 1->1->2, return 1->2.

   Given 1->1->2->3->3, return 1->2->3.

9. In a single/double linked list, user gives a certain range of index with shifting value. You need to right shift and rotate those range of values based on the given shifting value. Do the same for left shift.

   Given list: 2,9,6,3,**5,8,11,3,6,7**,13,5
   Given range: 4th index to 9th index(consider the starting index is 0) and shifting value is 2
   Output: 2,9,6,3,6,7,5,8,11,3,13,5

   Given list: 2,9,**6,3,5,8,11,3,6**,7,13,5
   Given range: 2nd index to 8th index(consider the starting index is 0) and shifting value is 4
   Output: 2,9,8,11,3,6,6,3,5,7,13,5
10. Reverse a linked list from position m to n. Do it in-place and in one-pass. For example: Given 1->2->3->4->5->NULL, m = 2 and n = 4, return 1->4->3->2->5->NULL.

11. Given a singly linked list

    L: L0 → L1 → … → Ln-1 → Ln,

    L0 → Ln → L1 → Ln-1 → L2 → Ln-2 → …

    You must do this in-place without altering the nodes' values.

    For example, Given {1,2,3,4,5,6}, reorder it to {1,6,2,5,3,4}.

12. Join two double circular linked list in the following manner.
    List 1: 12,13,14,15,16
    List 2: 40,50,60,71,72
    Resultant list: 12,40,13,50,14,60,15,71,16,72
13. Write a program to arrange all odd numbers first then all even numbers in a linked list.