Assignment

1. Given a singly linked list, write a function to swap elements pairwise.
Eg: 1→ 2→ 3→ 4→ 5→ 6→ 7→ 8 //Before
2→ 1→4→ 3→ 6→ 5→ 8→ 7 //After

2. Given two linked lists, insert nodes of second list into first list at alternate positions of first list.
Eg: First list is 5→7→17→13→11 and
Second is 12→10→2→4→6,
The resultant list should become 5→12→7→10→17→2→13→4→11→6

3. Given a linked list and two integers M and N. Perform the following:

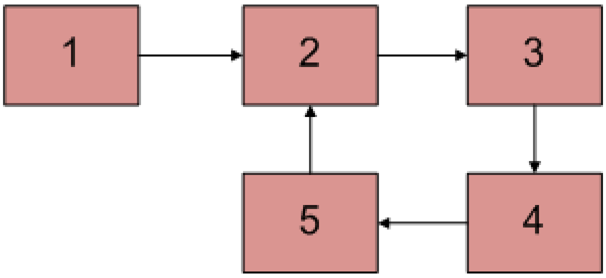
```
Eg:M = 2, N = 2
Linked List: 1->2->3->4->5->6->7->8
Output:
Linked List: 1->2->5->6
```

4. Given two numbers represented by two linked lists, write a function that returns sum list.

```
  First List: 5->6->3 // represents number 563
  Second List: 8->4->2 // represents number 842
Output
  Resultant list: 1->4->0->5 // represents number 1405
```

5. Given a singly linked list, rotate the linked list counter-clockwise by k nodes. Where k is a given positive integer.
Eg: Given linked list is
10→20→30→40→50→60 and k is 4,
The list should be modified to 50→60→10→20→30→40

6. Write a function *detectAndRemoveLoop()* that checks whether a given Linked List contains loop and if loop is present then it returns true.



.

7. Given a Linked List of integers, write a function to modify the linked list such that all even numbers appear before all the odd numbers in the modified linked list

Eg: Input: 17→15→8→12→10→5→4→1→7→6→NULL
Output: 8→12→10→4→6→17→15→5→1→7→NULL

Hint: The idea is to get pointer to the last node of list. And then traverse the list starting from the head node and move the odd valued nodes from their current position to end of the list.

8. Write a function to divide/split the circular linked list into two halves. The value from where the splitting has to be done will be given by the user.

9. Write a function to delete all the duplicate element form the list.
Eg: 1→ 2→ 3→ 3→ 4→ 5→ 6→ 6→ 2→ 7→ 8→ 6→ 9→ 10→ 9 → NULL //Before
1→ 2→ 3→ 4→ 5→ 6→ 7→ 8→ 9→ 10→ NULL //After

10. Write a function to find the number of count for any element in the linked list.
Eg: 1→ 2→ 3→ 3→ 4→ 5→ 6→ 6→ 2→ 7→ 8→ 6→ 9→ 10→ 9 → NULL //Before
Search for element 6.
OUTPUT: 3 times

11. Given an integer linked list of which both first half and second half are sorted independently. Write a function to merge the two parts to create one single sorted linked list in place.

Input 1: List 1:1→2→3→4→5→1→2; Output: 1→1→2→2→3→4→5

Input 2: 1→5→7→9→11→2→4→6; Output 2: 1→2→4→5→6→7→9→11

12. Write a linked list to reverse a single linked list.