**Module 22.5 Practice Day 1**

1 Given a singly linked list of characters, write a function that returns true if the given list is a palindrome, else false.

| Sample | Input | Output | Explanation |
| --- | --- | --- | --- |
| 1 | 5  R A D A R | True | The list is a Palindrome as if we reverse RADAR we will get the same word. |
| 2 | 8  FARANAAJ | False | Vice Versa to Sample 1. |

2 Given a singly linked list consisting of N nodes. The task is to remove duplicates (nodes with duplicate values) from the given list (if exist). Note that, The nodes are arranged in a sorted way.

Expected Time Complexity : O(N)

| Sample | Input | Output | Explanation |
| --- | --- | --- | --- |
| 1 | 4  2 2 4 5 | 2 -> 4 -> 5 | In the given linked list 2 ->2 -> 4-> 5, only 2 occurs more than 1 time. |
| 2 | 5  2 2 2 2 2 | 2 | In the given linked list 2 ->2 ->2 ->2 ->2, 2 is the only element and is repeated 5 times. |

3 Given a singly linked list, rotate the linked list counter-clockwise by k nodes. Where k is a given positive integer. For example, if the 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. Assume that k is smaller than the count of nodes in a linked list.

Expected Time Complexity : O(N)

| Sample | Input | Output | Explanation |
| --- | --- | --- | --- |
| 1 | 8  1 2 2 4 5 6 7 8  4 | 5->6->7->8->1->2->2->4 | 4th position Node is 4. So, from last till the 5th Position the rotation will happen. At first 8, then 7, then 6 and lastly 5 will go at the front to form 5->6->7->8->1->2->2->4 |
| 2 | 5  1 2 3 4 5  3 | 4->5->1->2->3 | 3rd position Node is 3. So, from last till the 4th Position the rotation will happen. At first 5, and lastly, 4 will go at the front to form 4->5->1->2->3 |

4 Given a singly linked list of size N. The task is to reverse every k node (where k is an input to the function) in the linked 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 (See Sample 2 for clarification). Your task is to complete the function **reverseKNodes()** which should reverse the linked list **in a group of size k** and return the **head of the modified linked list**.

Expected Time Complexity : O(N)

| Sample | Input | Output | Explanation |
| --- | --- | --- | --- |
| 1 | 8  1 2 2 4 5 6 7 8  4 | 4->2->2->1->8 ->7-> 6->5 | The first 4 elements 1,2,2,4 are reversed first and then the next 4 elements 5,6,7,8. Hence, the resultant linked list is 4->2->2->1->8->7->6->5. |
| 2 | 5  1 2 3 4 5  3 | 3->2->1->5->4 | The first 3 elements are 1,2,3 are reversed first and then elements 4,5 are reversed. Hence, the resultant linked list is 3->2->1->5->4. |

**Bonus Problem:**

Delete all of the elements from a Linear Linked List whose sum is equal to 0 (Zero).

Accepted Time Complexity : Upto O(n^2)

| Sample | Input | Output | Explanation |
| --- | --- | --- | --- |
| 1 | 8  6 -6 8 4 -12 9 8 -8 | 9 | 6-6 = 0  8+4-12 =0  8-8 =0  Thus, all of these numbers from the list is eliminated |
| 2 | 11  4 6 -10 8 9 10 -19 10 -18 20 25 | 20->25 | 4+6-10=0  8+10-18=0  9+10 = -19  Thus, all of these numbers from the list is eliminated |