**Week 5**

Realize the given problems using both singly and doubly linked list:

1. Given a linked list and two integers M and N. Traverse the linked list such that you retain M nodes then delete next N nodes, continue the same till end of the linked list.
2. Given two linked lists, insert nodes of second list into first list at alternate positions of first list.   
   For example, if first list is 5->7->17->13->11 and second is 12->10->2->4->6, the first list should become 5->12->7->10->17->2->13->4->11->6 and second list should become empty. The nodes of second list should only be inserted when there are positions available. For example, if the first list is 1->2->3 and second list is 4->5->6->7->8, then first list should become 1->4->2->5->3->6 and second list to 7->8.

Use of extra space is not allowed (Not allowed to create additional nodes), i.e., insertion must be done in-place. Expected time complexity is O(n) where n is number of nodes in first list.

1. Given a linked list of size N. The task is to reverse every k nodes (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.

Example 1:

Input:

LinkedList: 1->2->2->4->5->6->7->8

K = 4

Output: 4 2 2 1 8 7 6 5

Example 2:

Input:

LinkedList: 1->2->3->4->5

K = 3

Output: 3 2 1 5 4

Expected TimeComplexity: O(n)