Data Structures Assignment 1

1- Doubly Linked List implementation

1. insertAfter (data, afterData)
2. insertBefore (data, beforeData)
3. int count()
4. getDataByIndex(int index)
5. void reverse() – in place
6. *LinkedList reverse()

2- Remove every k-th node of the linked list

Given a linked list, the task is to remove every kth node of the linked list. Assume that k is always less than or equal to the length of the Linked List.

Create method that remove k-th node
Examples :

1- *Input: LinkedList: 1 -> 2 -> 3 -> 4 -> 5 -> 6, k = 2*
Output: 1 -> 3 -> 5
Explanation: After removing every 2nd node of the linked list, the resultant linked list will be: 1 -> 3 -> 5 .

2- *Input: LinkedList: 1 -> 2 -> 3 -> 4 -> 5 -> 6 -> 7 -> 8 -> 9 -> 10, k = 3*
Output: 1 -> 2 -> 4 -> 5 -> 7 -> 8 -> 10
Explanation: After removing every 3rd node of the linked list, the resultant linked list will be: 1 -> 2 -> 4 -> 5 -> 7 -> 8 -> 10.

3- Intersection of Two Sorted Linked Lists

Given two lists sorted in increasing order, create and return a new list representing the intersection of the two lists. The new list should be made with its own memory — the original lists should not be changed.

Create method that return new linked list with intersection of 2 linked lists

Example:

Input:

First linked list: 1->2->3->4->6

Second linked list be 2->4->6->8,

Output: 2->4->6.

The elements 2, 4, 6 are common in

both the list so they appear in the

intersection list.

Input:

First linked list: 1->2->3->4->5

Second linked list be 2->3->4,

Output: 2->3->4

The elements 2, 3, 4 are common in

both the list so they appear in the

intersection list.