**Module 19.5 Practice Day 2**

Q1 Implement the following Operations for **a Doubly Linked List** using the template code taught in Modules 17, 18, and 19.

1. Counting the Size
2. Display List
3. Insertion at Head
4. Insertion at Tail
5. Insertion at Specific Position
6. Search a value (Unique List)
7. Search a value (Duplication enabled List)
8. Insertion after a specific value (Unique List)
9. Insertion after a specific value (Duplication enabled List)
10. Deletion at Head
11. Deletion at Tail
12. Deletion at a Specific Position
13. Deletion by Value (Unique List)
14. Deletion by Value(Duplication enabled List)

Q2 What are Pass by Reference and Pass by value? Use the following two code snippets to explain.

| void insertatHead(Node\* &head, int val)  {  …….  ……  }  int main()  {  Node \*head = NULL;  insertatHead(head,5);  } | void insertatHead(Node\* head, int val)  {  …….  ……  }  int main()  {  Node head = NULL;  insertatHead(&head,5);  } |
| --- | --- |

Q3 Write down a program that will take n number of input from the users and create a Linear Linked List of n size. Replace all the even numbers in the list by -1 and display the List.

| Sample Input | Sample Output |
| --- | --- |
| 5  1 3 4 5 6 | 1-> 3 -> -1 -> 5 -> -1 |
| 6  2 2 2 1 2 3 | -1 -> -1->-1-> 1 -> -1 -> 3 |

Q4 Write down a function named **int findMid(Node\* &head)** which will return the value of middle element of the List without using **countLength()** function. [ Hints: Use two pointers to solve ]

Q5 Write down a program that will take n number of input from the users and create a Linear Linked List of n size. Now, write a Recursive function that will reverse till the kth position of a Linear Linked List. (k as user input).

**Input:**

* First line contains n and q
* Second line contains n space-separated integers denoting the elements of the Linear Linked List
* The next q line contains a single integer denoting the position k till which the List needs to reversed

**Output:**

q number of lines containing the reverse lists.

| Sample Input | Sample Output |
| --- | --- |
| 5 2  1 3 4 5 6  3  2 | 4->3->1  3->1 |
| 6 1  1 2 3 4 5 6  4 | 4->3->2->1 |