**Lab 7 Practice Exercise**

**Task 1:** Given a linked list and positions m and n. You need to reverse the linked list from position m to n. Reversed linked list should be separate from rest of list

**Examples:**

Input : linkedlist : 10->20->30->40->50->60->70->NULL , m = 3 and n = 6

Output : 10->20->60->50->40->30->70->NULL

Explanation: Linkedlist reversed starting from the node m i.e. 30 and n i.e. 60

**Task 2:** Given a singly linked list. The task is to remove duplicates (nodes with duplicate values) from the given list (if it exists).

Note: Try not to use extra space. The nodes are arranged in a sorted way.

**Example:**

Input: LinkedList: 2->2->4->5

Output: 2 -> 4 -> 5

**Task 3:** Take an array as input, which needs to be sorted according to the contents of the 2nd array. The user should input values for n (length of 1st array) and m (length of 2nd array) and the contents of both these arrays. The contents of the first array should be sorted according to the 2nd, without interrupting the order of the other elements. For example:

Input:

N = 6, arr1 = {7,13,4,7,10,4}

M = 2, arr2 = {7,4}

Output:

Arr1 = {7,7,4,4,13,10}

According to the contents of the 2nd array, all 7’s should come first. Then all 4’s.. And the order of the other elements should not be disturbed.

**Task 4:** Given a stack, the task is to reverse the stack using the queue data structure.

Examples:

Input: Stack: (Top to Bottom) [10 -> 20 -> 30 -> 40]

Output: Stack: (Top to Bottom) [40 -> 30 -> 20 -> 10]

Input: Stack: [6 -> 5 -> 4]

Output: Stack: [4 -> 5 -> 6]

**Task 5:** Given a stack with push(), pop(), and empty() operations, The task is to delete the middle element of it without using any additional data structure.

Input : Stack[] = [1, 2, 3, 4, 5]

Output : Stack[] = [1, 2, 4, 5]

Input : Stack[] = [1, 2, 3, 4, 5, 6]

Output : Stack[] = [1, 2, 4, 5, 6]

**Task 6:** Provide the code to convert the “**((A + B) – C \* (D / E)) + F”** expression into post fix using stack.

Input: ((A + B) – C \* (D / E)) + F

Output: AB+CDE/\*-F+

**Task 7:** Write a program to print the numbers from 1 to 10 in such a way that when number is odd increment 1, if number is even subtract 1. Attempt this question using recursion.

Output: 2, 1, 4, 3, 6, 5, 8, 7, 10, 9

**Task 8:**

1. Generate the following sequence with recursive approach
2. , 3 , 6 , 10 , 15 , 21 , 28 . . . .
3. Generate the following sequence with recursive approach

0 , 1 , 1 , 2 , 3 , 5 , 8 , 13 , 21 , 34 , 55 , 89 , 144 . . .

**Task 9:** Write a recursive function that takes two integer as a parameters and finds out all composite and prime number in given range.