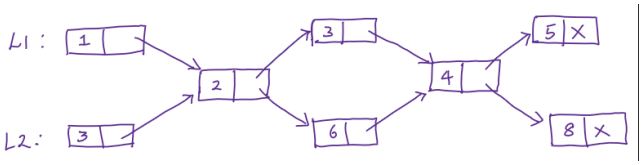
6th Lab – Practice Lab – Arrays, Stacks Queues and Linked Lists

1. Let Q be a queue with n positive integers. Write a C Program to create two new queues Q1 and Q2 such that all even numbers of Q are enqueued in Q1 and all odd numbers of Q are enqueued in Q2.
2. Let S be a stack with n positive integers. Write a C Program to display multiple of 3 in the stack S and also illustrates your algorithm for any sample input. Compute the running time of your algorithm.
3. Let L1 and L2 be two Singly-linked lists. Write a C Program to output $L1 \cap L2$.
4. Let L1 and L2 be two linked lists with 'n' elements each. Let k be a positive integer, with k<n. Write a C Program that will re-direct the pointers at the k^{th} position such that the elements coming after the k^{th} position in L1 and L2, now belong to L2 and L1 respectively. You should not exchange the data elements, but only write the necessary pseudo-code to alter the pointers. For eg. If L1: 1 2 3 4 5 and L2: 6 7 8 9 10 and k=3, then the new contents of L1 and L2 are, L1:1 2 3 9 10 and L2: 6 7 8 4 5. Implement the code for both Single and Doubly linked lists
5. Let L be a linked list and k be a positive integer. Write a C Program to split L at the position k such that elements appearing before k(including k) are all even numbers and elements appearing after k(excluding k) are all odd numbers. If the count of even numbers out-number the count of odd numbers, append the remaining even numbers at the end of the list. Implement the code for both Single and Doubly linked lists
6. Write a C Program, which reads 'n' numbers and computes the sum of all the predecessors of every even number that occurs among the given n numbers. The Predecessor of a number is the number that occurs before that number. For example, if n=7 and the given numbers are 17, 11, 24, 13, 7, 8, 14 then the answer is (11+7+8=26) .
7. Let A and B be arrays of n integers each. A common subsequence of A and B is a sequence(not necessarily contiguous) of integers of A that also belongs to B. For example, if the entries of A are 5, 8, 6, 4, 7, 1, 3 and the entries of B are 4, 5, 6, 9, 7, 3, 2, a common subsequence is 5, 6, 7. Write an algorithm to find a common subsequence of A and B. Compute the running time of your algorithm.
8. Let S be a stack with n positive integers. Write an algorithm to create two new stacks S1 and S2 such that all even numbers of S are added into S1 and all odd numbers of S are added into S2. Illustrate your algorithm for any sample input and output the total number of push( ) and pop( ) operations that are performed in the algorithm.
9. Let Q be a queue with n positive integers. Write an algorithm to display multiple of 3 in the queue Q and also illustrates your algorithm for any sample input. Compute running time of your algorithm.
10. Let L1 and L2 be two Singly linked lists. Write an algorithm to output $L1 \cup L2$.
11. Let L1 and L2 be two linked lists with 'n' elements each. Define a condition C as follows: Merge both the lists at a node k to form a single common node if the kth node of both L1 and L2 contain the same element. Do this for all nodes in both the lists if the condition C is satisfied. Implement the code for both Single and Doubly linked lists
    Note: Define the linked list accordingly so as to validate the necessary conditions.
    For e.g. if L1: 1 2 3 4 5 and L2: 3 2 6 4 8 Then the merged linked list would look like,



12. Let L be a linked list and k be a positive integer. Write an algorithm to split L at the position k such that elements appearing before k(including k) are all odd numbers and elements appearing after k(excluding k) are all even numbers. If the count of even numbers out-number the count of odd numbers, append the remaining even numbers at the end of the list. Implement the code for both Single and Doubly linked lists
13. Write an algorithm, which reads 'n' numbers and computes the sum of all the successors of every odd numbers that occur among the given n numbers. Successor of a number is the number that occurs after that number in the array. For example, if n=7 and the given numbers are 17, 11, 24, 13, 7, 8, 14 then answer is (11+24+7+8=50) .
14. **Let A be a given string. A palindrome** is a string when it reads the same backward as forward. A **substring** is a palindrome if a contiguous sequence of characters within the string is a palindrome. Write an algorithm that will identify a sub-string of A that is a palindrome. Derive the time complexity of your algorithm.