

Non-linear Data Structures & Problem Solving (20B16CS324)

Lab Assignment 1

1. Given a singly linked list, write a program to swap elements pairwise.

Eg: $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8$ //Before

$2 \rightarrow 1 \rightarrow 4 \rightarrow 3 \rightarrow 6 \rightarrow 5 \rightarrow 8 \rightarrow 7$ //After

2. Given two linked lists, write a program to insert nodes of second list into first list at alternate positions of first list.

Eg: First list is $5 \rightarrow 7 \rightarrow 17 \rightarrow 13 \rightarrow 11$

Second is $12 \rightarrow 10 \rightarrow 2 \rightarrow 4 \rightarrow 6$

The resultant list should become $5 \rightarrow 12 \rightarrow 7 \rightarrow 10 \rightarrow 17 \rightarrow 2 \rightarrow 13 \rightarrow 4 \rightarrow 11 \rightarrow 6$

3. Given two numbers represented by two linked lists, write a program that returns sum list.

Eg: First List: $5 \rightarrow 6 \rightarrow 3$ // represents number 563

Second List: $8 \rightarrow 4 \rightarrow 2$ // represents number 842

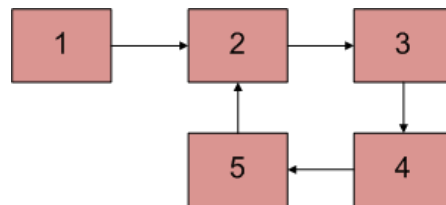
Resultant list: $1 \rightarrow 4 \rightarrow 0 \rightarrow 5$ // represents number 1405

4. Given a singly linked list, rotate the linked list counter-clockwise by k nodes. Where k is a given positive integer.

Eg: Given linked list is $10 \rightarrow 20 \rightarrow 30 \rightarrow 40 \rightarrow 50 \rightarrow 60$ and k is 4

The list should be modified to $50 \rightarrow 60 \rightarrow 10 \rightarrow 20 \rightarrow 30 \rightarrow 40$

5. Write a function *detectAndRemoveLoop()* that checks whether a given Linked List contains loop and if loop is present then it returns true.



6. Given a Linked List of integers, write a program to modify the linked list such that all even numbers appear before all the odd numbers in the modified linked list.

Eg: Input: $17 \rightarrow 15 \rightarrow 8 \rightarrow 12 \rightarrow 10 \rightarrow 5 \rightarrow 4 \rightarrow 1 \rightarrow 7 \rightarrow 6 \rightarrow \text{NULL}$

Output: $8 \rightarrow 12 \rightarrow 10 \rightarrow 4 \rightarrow 6 \rightarrow 17 \rightarrow 15 \rightarrow 5 \rightarrow 1 \rightarrow 7 \rightarrow \text{NULL}$

Hint: The idea is to get pointer to the last node of list. And then traverse the list starting from the headnode and move the odd valued nodes from their current position to end of the list.

7. Write a program to divide/split the circular linked list into two halves. The value from where the splitting has to be done will be given by the user.

8. Write a program to delete all the duplicate element form the list.

Eg: $1 \rightarrow 2 \rightarrow 3 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 6 \rightarrow 2 \rightarrow 7 \rightarrow 8 \rightarrow 6 \rightarrow 9 \rightarrow 10 \rightarrow 9 \rightarrow \text{NULL}$ //Before

$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 10 \rightarrow \text{NULL}$ //After

9. Write a program to find the number of count for any element in the linked list.

Eg: $1 \rightarrow 2 \rightarrow 3 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 6 \rightarrow 2 \rightarrow 7 \rightarrow 8 \rightarrow 6 \rightarrow 9 \rightarrow 10 \rightarrow 9 \rightarrow \text{NULL}$ (Search for element 6)

OUTPUT: 3 times

10. Given an integer linked list of which both first half and second half are sorted independently. Write a program to merge the two parts to create one single sorted linked list in place.

Eg: Input: a: 5->10->15, b: 2->3->20

Output: 2->3->5->10->15->20

11. Write a program to reverse a double linked list.