SINGLY LINKED LIST

Linked List Insertion,

Linked List Deletion,

How to write C functions that **modify head pointer** of a Linked List?

Write a function to get N^{th} node in a Linked List,

N^{th} node from the end of a Linked List,

Write a function that counts the number of times a given int occurs in a Linked List,

Given a linked list which is sorted, how will you insert in sorted way,

Print the middle of a given linked list,

DELETION/REMOVAL:

Write a function to delete a Linked List,

Given only a pointer to a node to be deleted in a singly linked list, how do you delete it?

Delete nodes which have a greater value on right side,

Delete alternate nodes of a Linked List,

Delete N nodes after M nodes of a linked list,

Delete a given node in Linked List under given constraints,

Remove duplicates from a sorted linked list,

Remove duplicates from an **unsorted** linked list,

Given a linked list of line segments, remove middle points,

ROTATION/SWAPPING/REVERSAL:

Rotate a Linked List,

Reverse a linked list,

Recursive function to print reverse of a Linked List,

Reverse a Linked List in groups of given size,

Reverse alternate K nodes in a Singly Linked List,

Pairwise swap elements of a given linked list,

Pairwise swap elements of a given linked list by changing links,

SINGLY LINKED LIST

Function to check if a singly linked list is palindrome,

Identical Linked Lists,

Intersection of two Sorted Linked Lists,

Intersection point of two Linked Lists.,

Detect loop in a linked list,

Detect and **Remove Loop** in a Linked List,

Merge two sorted linked lists,

Merge Sort for Linked Lists,

Merge a linked list into another linked list at alternate positions,

Sort a linked list of 0s, 1s and 2s,

QuickSort on Singly Linked List,

Alternating split of a given Singly Linked List,

Move last element to front of a given Linked List,

Segregate even and odd nodes in a Linked List,

Flattening a Linked List,

Flatten a multilevel linked list

Union and Intersection of two Linked Lists,

Find a triplet from three linked lists with **sum** equal to a given number,

Add two numbers represented by linked lists | Set 1,

Add two numbers represented by linked lists | Set 2,

Construct a Maximum Sum Linked List out of two Sorted Linked Lists having some Common nodes,