Program to implement factorial of given number

Program to implement Fibonacci Series

Program to implement Tower of Hanoi

Program to perform various operations on Singly Linked List with one external reference pointer

Program to perform various operations on Singly Linked List with two external reference pointers

Program to implement various operations on doubly linked list

Program to implement various operations on circular linked list

Remove duplicate nodes from a sorted linked list

For example, if the linked list contains 1,2,2,2,3,4,4,5 then output shall be 1,2,3,4,5

Construct a linked list by merging alternate nodes of two linked list

For example, if given linked list are 1,9,3 and 6,11,5 then output list shall be 1,6,9,11,3,5

Move even nodes of the linked list at the end of linked list in reverse order

For example, if the linked list is 1,3,2,6,8,9,11 then output list shall be 1,2,8,11,9,6,3

Check if given linked list is a palindrome

For example, 1,3,7,3,1 list is a palindrome

We are given a linked list, we need to print the linked list in reverse order.

Examples:

Input : list : 5-> 15-> 20-> 25

Output : Reversed Linked list : 25-> 20-> 15-> 5

Input : list : 85-> 15-> 4-> 20

Output : Reversed Linked list : 20-> 4-> 15-> 85

Input : list : 85

Output : Reversed Linked list : 85

### 2:

Given two linked lists, the task is to check whether the first list is present in 2nd list or not.

Input : list1 = 10->20

list2 = 5->10->20

Output : LIST FOUND

Input : list1 = 1->2->3->4

list2 = 1->2->1->2->3->4

Output : LIST FOUND

Input : list1 = 1->2->3->4

list2 = 1->2->2->1->2->3

Output : LIST NOT FOUND

### 3:

Given a sorted linked list and a value to insert, write a function to insert the value in sorted way.

Initial Linked List  
SortedLinked List

Linked List after insertion of 9  
UpdatedSortedLinked List

### 4:

Given a singly linked list and a number k, find the last node whose n%k == 0, where n is the number of nodes in the list.

Examples:

Input : list = 1->2->3->4->5->6->7

k = 3

Output : 6

Input : list = 3->7->1->9->8

k = 2

Output : 9

Implement stack using array

Implement infix to prefix

Implement infix to postfix

### 

### 1:

Check if given expression is balanced expression or not

Given a string containing opening and closing braces, check if it represents a balanced expression or not.

For example, { [ { } { } ] } , { { [ ] } }  are balanced expressions

{ [ } ] , { { [ ] { } ] }  are not balanced expressions.

### 2:

Evaluate given postfix expression

For example,

82/ will evaluate to 4  (8/2)

138\*+ will evaluate to 25 ( 1+8\*3)

545\*+5/ will evaluate to 5  ((5+4\*5)/5)

### 3:

Decode a given sequence to construct minimum number in the series without using repeated digits    ( I denotes Increase and D denotes Decrease in the sequence

Sequence   : IIDDIDID

Output: 125437698

Sequence:  IDIDII

Output:   1325467

### 

### 4:

Given an array, print the Next Greater Element (NGE) for every element. The Next greater Element for an element x is the first greater element on the right side of x in array. Elements for which no greater element exist, consider next greater element as -1.

Examples:  
a) For any array, rightmost element always has next greater element as -1.  
b) For an array which is sorted in decreasing order, all elements have next greater element as -1.  
c) For the input array [4, 5, 2, 25}, the next greater elements for each element are as follows.

Element       NGE

   4      -->   5

   5      -->   25

   2      -->   25

   25     -->   -1

d) For the input array [13, 7, 6, 12}, the next greater elements for each element are as follows.

  Element        NGE

   13      -->    -1

   7       -->     12

   6       -->     12

   12     -->     -1

## **Common Problems**

Queue using array

Circular Queue using array

Priority Queue using array

### 

### 1:

Suppose there is a circle. There are n petrol pumps on that circle. You are given two sets of data.

1. The amount of petrol that every petrol pump has.  
2. Distance from that petrol pump to the next petrol pump.

Calculate the first point from where a truck will be able to complete the circle (The truck will stop at each petrol pump and it has infinite capacity). Expected time complexity is O(n). Assume for 1 litre petrol, the truck can go 1 unit of distance.

For example, let there be 4 petrol pumps with amount of petrol and distance to next petrol pump value pairs as {4, 6}, {6, 5}, {7, 3} and {4, 5}. The first point from where truck can make a circular tour is 2nd petrol pump. Output should be “start = 1” (index of 2nd petrol pump).

### 2:

Given a number n, write a function that generates and prints all binary numbers with decimal values from 1 to n.

Examples:

Input: n = 2

Output: 1, 10

Input: n = 5

Output: 1, 10, 11, 100, 101

### 

### 

### 4:

Given an array and a positive integer k, find the first negative integer for each and every window(contiguous subarray) of size k. If a window does not contain a negative integer, then print 0 for that window.

Examples:

Input : arr[] = {-8, 2, 3, -6, 10}, k = 2

Output : -8 0 -6 -6

First negative integer for each window of size k

{-8, 2} = -8

{2, 3} = 0 (does not contain a negative integer)

{3, -6} = -6

{-6, 10} = -6

Input : arr[] = {12, -1, -7, 8, -15, 30, 16, 28} , k = 3

Output : -1 -1 -7 -15 -15 0

## **Common Problems**

Implement stack using linked list

Implement queue using linked list

## **Common Problems**

Program to implement linear search

Program to implement binary search

Program to develop computer game for guessing number between 1-100

## **Common Problems**

1.      Program to implement bubble Sort

2.      Program to implement selection sort

3.      Program to implement insertion sort

## **Common Problems**

1.      Program to perform Merge Sort

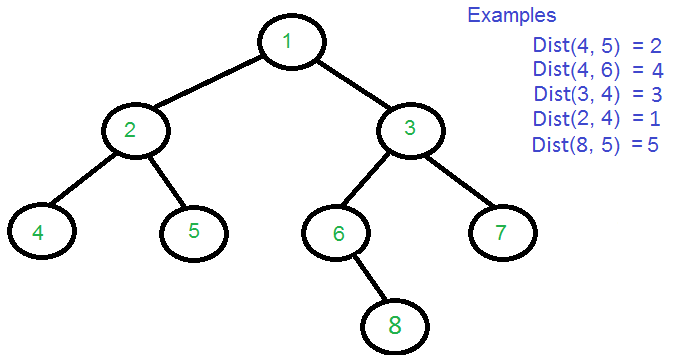
2.      Program to perform Quick Sort

## Common Problems:

Program to implement Binary Search Tree

### Batch 1 :

Find the distance between two keys in a binary tree, no parent pointers are given. Distance between two nodes is the minimum number of edges to be traversed to reach one node from other.

[](https://www.geeksforgeeks.org/wp-content/uploads/dist.png)

### 

### Batch 2:

Given a Binary Tree, find sum of all left leaves in it. For example, sum of all left leaves in below Binary Tree is 5+1=6.

tree

### 

### Batch 3:

Given a Binary Tree, write a function to check whether the given Binary Tree is a prefect Binary Tree or not.

A Binary tree is Perfect Binary Tree in which all internal nodes have two children and all leaves are at same level.

Examples:  
The following tree is a perfect binary tree

10

/ \

20 30

/ \ / \

40 50 60 70

18

/ \

15 30

The following tree is not a perfect binary tree

1

/ \

2 3

\ / \

4 5 6

### Batch 4:

Given a binary tree, task is to find sub tree with maximum sum in tree.

Examples:

Input: 1

/ \

2 3

/ \ / \

4 5 6 7

Output: 28

As the entire tree elements are positive,

the largest subtree sum is equal to

sum of all tree elements.

Input : 1

/ \

-2 3

Program to represent graph using adjacency list and traverse it