

## List of Practice Programs

### Linked Lists:

1. Implement linked list ADT with the following functions.
  - a. Insertion
  - b. Deletion
  - c. Display
2. Write a program to reverse a singly linked list.
3. Write a program to detect whether a loop exists in the linked list. The program should also print the starting node of the loop.
4. Write a program to merge two linked lists alternatively.
5. Write a program to merge two sorted linked lists into a single sorted list.
6. Write a program to reverse a singly linked list in K groups:  
Given a linked list, write a function to reverse every k nodes (where k is an input to the function).  
**Input:** 1->2->3->4->5->6->7->8->NULL, K = 3  
**Output:** 3->2->1->6->5->4->8->7->NULL  
**Input:** 1->2->3->4->5->6->7->8->NULL, K = 5  
**Output:** 5->4->3->2->1->8->7->6->NULL
7. Write a program to sort the elements in a singly linked list using selection sort.

### Stack:

1. Implement stack ADT with the following functions using arrays
  - a) Push (a)
  - b) Pop ()
  - c) Peek ()
  - d) Display
2. Implement stack ADT with the following functions using Linked Lists  
Push (a)  
Pop ()  
Peek ()  
Display
3. Write a program to evaluate a post fix expression
4. Write a program to convert an infix expression to a post fix expression
5. Next Greater Frequency Element: Given an array, for each element find the value of the nearest element to the right which is having a frequency greater than as that of the current element. If there does not exist an answer for a position, then make the value '-1'.  
Example: 1) Input : a[] = [1, 1, 2, 3, 4, 2, 1]  
Output : [-1, -1, 1, 2, 2, 1, -1]  
2) Input : a[] = [1, 1, 1, 2, 2, 2, 2, 11, 3, 3]  
Output : [2, 2, 2, -1, -1, -1, -1, 3, -1, -1]
6. Next Greater element: 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 the array. Elements for which no greater element exist, consider the next greater element as -1.

Examples: a) For an array, the rightmost element always has the next greater element as -1.

- b) For an array that is sorted in decreasing order, all elements have the next greater element as -1.
- c) For the input array [4, 5, 2, 25]  
Output is [5, 25, 25, -1]
- e) Input: [13, 7, 6, 12]  
Output: [-1, 12, 12, -1]

Queue:

1. Write a program to implement Queue ADT with the following operations using Arrays
  - a. Enqueue( a)
  - b. Dequeue()
  - c. Display()
  - d. Front()
  - e. Rear()
2. Write a program to implement Queue ADT with the following operations using Linked Lists
  - a. Enqueue( a)
  - b. Dequeue()
  - c. Display()
  - d. Front()
  - e. Rear()
3. Implement a stack using the queue data structure.
4. Implement a queue using the stack data structure.