**Task-1:**

* Insert 10 Integers values in the stack
* If the Insert input reach the Highest index of Array display the message Stack overflow
* Remove the Inserted values till the Last value and print the message that stack is empty

**Task-2:**

* Insert 10 Integers values in the stack
* Write a utility function for upper code to display all the inserted integer values in the linked list in forward and reverse direction both
* Write utility function to pop top element from the stack

**Task-4:**

1. Write a function **QueueCapacity** when the Queue is Full
2. Write a function **ADDMember** when a new integer value is added in the array
3. Write a function **RemoveMember** when any data member is remove from the queue

**Task-5:**

Use the Upper code snippet implement the following utility function in the Link based Queue

* Write a function **QueueCapacity** when the Queue is Full
* Write a function **ADDMember** when a new integer value is added in the linkedlist
* Write a function **RemoveMember** when any data member is remove from the queue

**Task – 6:**

Write a function to construct a binary tree

* 2. Data must be provided by user at the run time
* 3. -1 input from user indicate the is no child for particular node
* 4. Print the Binary tree in format of “Preorder, Inorder, Postorder”

**Task-7:**

1. Use the Upper code snippet implement the utility function with the help of array based stack **infixToPostfix** by using sample pseudocode.