**Data structures Lab Task II/IV B.Tech**

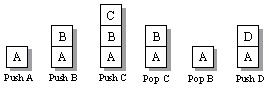
Batch-1: 07-December-2021 @ 9.30 to 12

Batch-2: 09-December-2021 @ 8.40 to 11.10

**Task-1**

We have discussed stack operations called push () and pop () in the class. Also illustrated with examples. Design and implement menu driven C program with 4 operations. (1) Add element to stack (2) Delete element from stack (3) Traverse elements and (4) Exit. Write all possible examples supported by relevant test cases.

Sample test case is as below:



**Task-2**

Design and implement C program to check a string is palindrome or not using a stack. A **stack** is LAST IN FIRST OUT (LIFO) data structure. The element which is inserted last, is accessed first. Insertion and deletion of elements happens only at top of the **Stack**. The sequence of exit of elements from a stack is reverse of the sequence of their entry in stack.

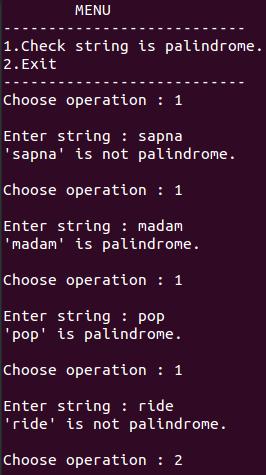
Sequence of Entry.  
A --> B --> C -- > D --> E  
Sequence of Exit.  
E --> D --> C --> B --> A

**Algorithm to check palindrome string using stack**

* Find the length of the input string using strlen function and store it in a integer variable "length".
* Using a for loop, traverse input string from index 0 to length-1 and push all characters in stack.
* Remove (Pop) characters from stack one by one using a for loop and compare it with corresponding character of input string from beginning(traverse from index 0 to length-1). If we found a mismatch the input string is not a palindrome string otherwise **palindrome string**.

Write all possible examples supported by relevant test cases.

Sample test case is as below:

****

Task-3

You have an empty sequence, and you will be given

queries. Each query is one of these three types:

1 x -Push the element x into the stack.

2 -Delete the element present at the top of the stack.

3 -Print the maximum element in the stack.

**Function Description**

Complete the *getMax* function in the editor below.

*getMax* has the following parameters:   
- *string operations[n]:* operations as strings

**Returns**   
- *int[]:* the answers to each type 3 query

**Input Format**

The first line of input contains an integer, The next lines each contain an above mentioned query.

**Constraints**   
  
All queries are valid.

**Sample Input**

STDIN Function

----- --------

10 operations[] size n = 10

1 97 operations = ['1 97', '2', '1 20', ....]

2

1 20

2

1 26

1 20

2

3

1 91

3

**Sample Output**

26

91