Assignment Number: 9

Problem statement:

String operations: A palindrome is a string of characters that's identical when read in forward and backward direction. Typically, punctuation, capitalization, and spaces are ignored. For example, “1.Poor Dan is in a droop!!” is a palindrome, as can be seen by examining the characters “poordanisinadroop” and observing that they are identical when read forward and backward directions. One way to check for a palindrome is to reverse the characters in the string and compare them with the original-in a palindrome, the sequence will be identical.
Write C++ program with functions using Standard Template Library (STL) stack-
1. To print original string followed by reversed string using stack
2. To check whether given string is palindrome or not

Objectives:

- Understand the use of stack
- To know the applications of stack

Theory:

STL : In C++, the Standard Template Library (STL) provides a set of programming tools to implement algorithms and data structures like vectors, lists, queues, etc.

STL implements these data structures and algorithms using general-purpose classes and functions that have been tested rigorously.

C++ STL has 3 major components:

- Containers
- Iterators
- Algorithms

In addition to these, STL also provides several other features, including function objects, smart pointers, and exception handling mechanisms.

Stack is a controlled linear data structure which follows a particular order in which the operations are performed. The order may be LIFO(Last In First Out) or FILO(First In Last Out).

Operations on stack:

- **Push:** Adds an item in the stack. If the stack is full, then it is said to be an Overflow condition.
- **Pop:** Removes an item from the stack. The items are popped in the reversed order in which they are pushed. If the stack is empty, then it is said to be an Underflow condition.
- **getTop**: Returns top element of stack.

- **isEmpty:** Returns true if stack is empty, else false.
- **isFull:** Returns true if stack is full, else false.

There are two ways to implement a stack:

- Using array
- Using linked list

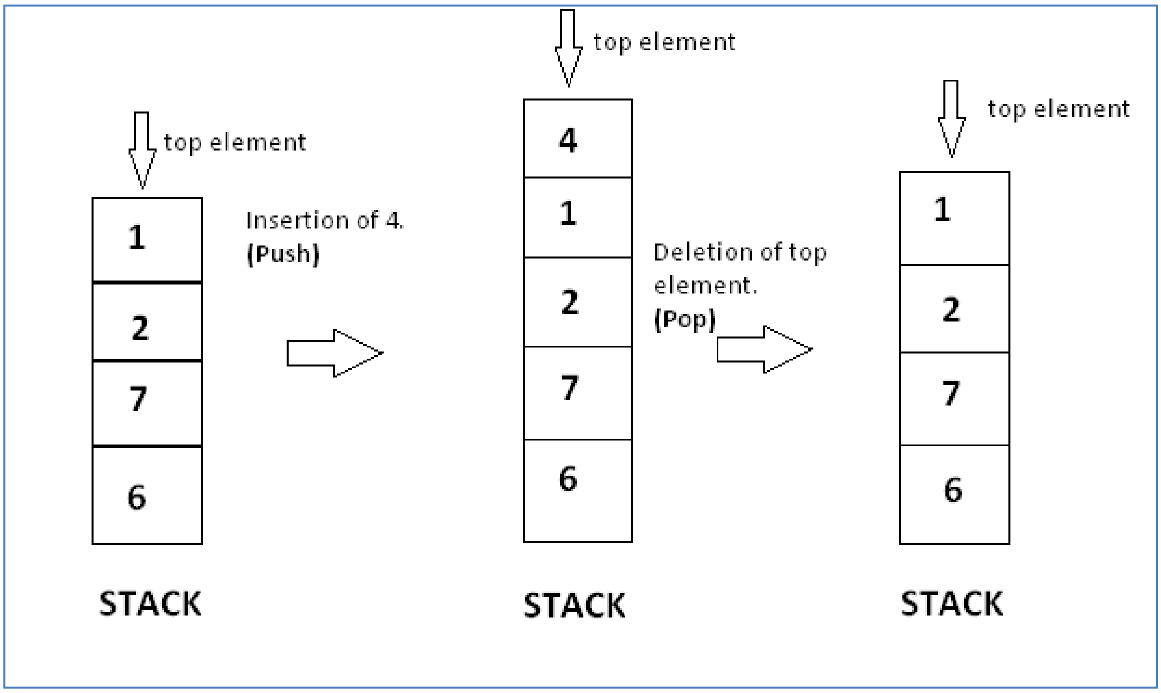

Fig. Graphical representation of stack

Features of stack implementation using Linked List:

- Dynamic data structures
- Do not have a fixed size
- Do not consume a fixed amount of memory

Algorithm:

For reverse the string :

Input: Input string str1[ ] as “poor dan is in a droop”

Output: Reverse string as str2[ ] “poordanisinadroop”

Step 1: Start

Step 2: Read input string

Step 3: for i=0 to str1[i]!=’\0’

```
                    If it is character then push each character into stack

                    If it is space then ignore it

Step 3: for i=0 to till stack get empty

        Pop each character from stack and copy into str2[i]


Step 4: Display reverse string

Step 5: Stop
```

Time complexity:

push() ______________

pop()________________

isEmpty()____________

getTop().____________

Conclusion: Thus we have successfully implemented string operation reverse and palindrome by using stack

Test cases
For string reverse and palindrome

1. madam
2. Madam
3. “Madam”
4. 1.Poor Dan is in a droop!

Practice problem:

Write code to convert decimal number in to binary number and write its time complexity.