Page No.
Date

Assignment No. - 09

Sumit Gulab Bhamare
SEComp A08
Sub - DSL

Aim: To illustrate the concept of stack & string.

Problem Statement: A palindrome is a string of character that's the same forward & backward. Typically, punctuation, capitalization, & spaces are ignored. For example, 'Poor Dan is in a droop' is a palindrome, as can be seen by examining the characters -'poor danisina droop' & ~~observe~~ observing that they are the same forward & backward. One way to check for a palindrome is to reverse the characters in the string & then compare with the original.

In a palindrome, the sequence will be identical. Write C++ program with functions-

1. To check whether given string is palindrome or not that uses a stack to determine whether a string is a palindrome.
2. To remove spaces & punctuation in string, convert all the characters to lowercase, & then call above Palindrome checking function to check for a palindrome.
3. To print string in reverse order using stack

Learning Objectives: To understand concept of stack & string.
To analyze the string is palindrome or not.

Learning Outcome: Students will be able to implement stack & queue data structures & algorithms for solving different kinds of problems.

Page No.
Date

Theory:
Stack:

It is named stack as it behaves like a real-world stack, for example - deck of cards or pile of plates etc. A stack is an abstract data type that serves as a collection of elements, with two principal operations: push, which adds an element to the collection, & pop, which removes the most recently added element that was not yet removed. The order in which elements come off a stack gives rise to its alternative name, LIFO (for last in, first out). Additionally, a peek operation may give access to the top without modifying the stack.

A real-world stack allows operations at one end only. For example, we can place or remove a card or plate from top of the stack only. Likewise, Stack ADT allows all data operations at one end only. At any given time, we can only access the top element of a stack.

This feature makes it LIFO data structure. LIFO stands for Last-in-first-out. Here, the element which is placed (inserted or added) last, is accessed first. In stack terminology, insertion operation is called PUSH operation & removal operation is called POP operation.

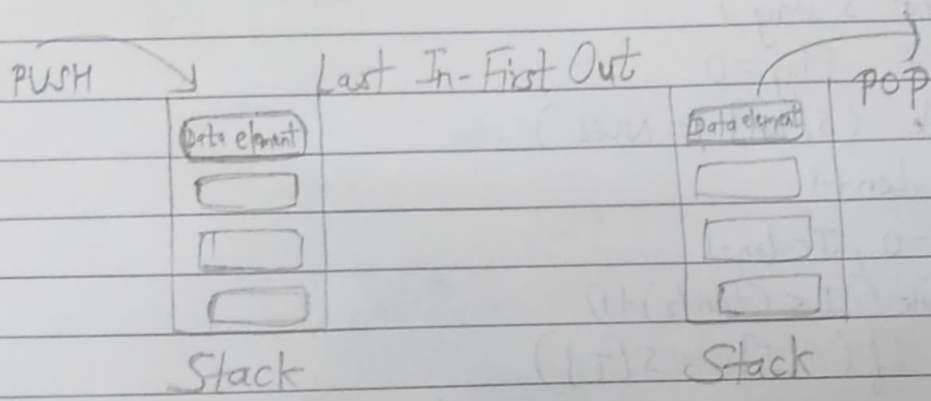
Stack operations may involve initializing the stack, using it & then de-initializing it. Apart from these basic stuffs, a stack is used for the following two primary operations -

push() - pushing (storing) an element on the stack
pop() - removing (accessing) an element from the stack

Page No.
Date

Stack Representation:

Below given diagram tries to depict a stack & its operations-



When data is PUSHed onto stack.

To use a stack efficiently we need to check status of stack as well. For the same purpose, the following functionality is added to stacks -

- peek() - get the top data element of the stack, without removing it.
- ~~isFull()~~
- isFull() - check if stack is full.
- isEmpty() - check if stack is empty.

At all times, we maintain a pointer to the last PUSHed data on the stack. As this pointer always represents the top of the stack, hence named top. The topPointer provides top value of the stack without actually removing it.

Input: Enter the string.

Output: Entered string is palindrome or not.

Page No.
Date

Algorithm:

Step 1: Input S(string)
Step 2: Len=0, Flag=0
Step 3: While (S[Len] != NULL)
        Len++
Step 4: I=0, J=Len-1
Step 5: While (I<(Len/2)+1)
        If (S[I]== S[J])
            Flag=0
        else
            Flag=1
        I++, J--
Step 6:
        If (Flag==0)
            Print Key Is a Palindrome
        else
            Print Key Is Not a Palindrome

Step 7: End

Software required: g++/gcc compiler-/ 64 bit fedora

Conclusion: Thus we have studied the implementation of string is palindrome or not.