Problem Statement: A palindrome is a string of character that‘s the same forward and backward. Typically, punctuation, capitalization, and 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” and observing that they are the same forward and backward. One way to check for a palindrome is to reverse the characters in the string and then compare with them the original-in a palindrome, the sequence will be identical. Write C++ program with functions- 1. To print original string followed by reversed string using stack 2. To check whether given string is palindrome or not  
  
  
#include <iostream>

#include <stack>

#include <cctype>

#include <string>

using namespace std;

string cleanString(const string& str) {

string cleaned = "";

for (char ch : str) {

if (isalnum(ch)) {

cleaned += tolower(ch);

}

}

return cleaned;

}

string reverseString(const string& str) {

stack<char> s;

string reversed = "";

for (char ch : str) {

s.push(ch);

}

while (!s.empty()) {

reversed += s.top();

s.pop();

}

return reversed;

}

bool isPalindrome(const string& str) {

string cleaned = cleanString(str);

string reversed = reverseString(cleaned);

return cleaned == reversed;

}

int main() {

string input;

cout << "Enter a string: ";

getline(cin, input); // Read the entire line of input

cout << "Original String: " << input << endl;

string reversedInput = reverseString(input);

cout << "Reversed String: " << reversedInput << endl;

if (isPalindrome(input)) {

cout << "The string is a palindrome." << endl;

} else {

cout << "The string is not a palindrome." << endl;

}

return 0;

}