```
#include <iostream>
#include <stack>
#include <cctype>
using namespace std;
void printOriginalAndReversed(const string& str) {
    cout << "Original String: " << str << endl;
    stack<char> s;
    for (int i = 0; i < str.length(); i++) {
        s.push(str[i]);
    }
    cout << "Reversed String: ";
    while (!s.empty()) {
        cout << s.top();
        s.pop();
    }
    cout << endl;
}
bool isPalindrome(const string& str) {
    int left = 0;
    int right = str.length() - 1;
    while (left < right) {
                if (!isalnum(str[left])) {
            left++;
        } else if (!isalnum(str[right])) {
            right--;
        } else {
            if (tolower(str[left]) != tolower(str[right])) {
                return false;
            }
            left++;
            right--;
        }
    }
```

```
        return true;
}
int main() {
    string input;
    cout << "Enter a string: ";
    getline(cin, input);
    printOriginalAndReversed(input);
    if (isPalindrome(input)) {
        cout << "The string is a palindrome." << endl;
    } else {
        cout << "The string is NOT a palindrome." << endl;
    }
    return 0;
}
```

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-

a)To print original string followed by reversed string using stack

b)To check whether given string is palindrome or not