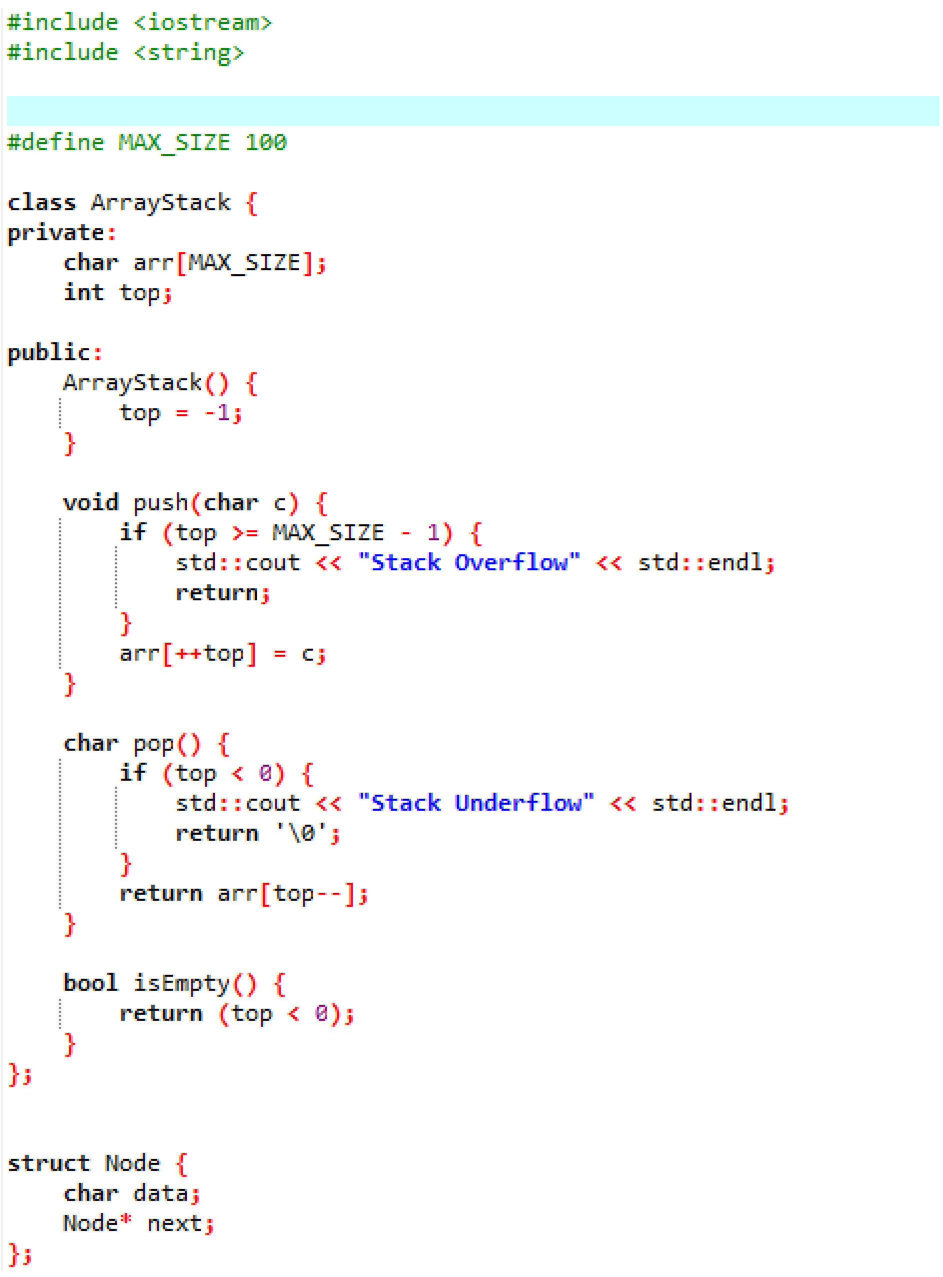
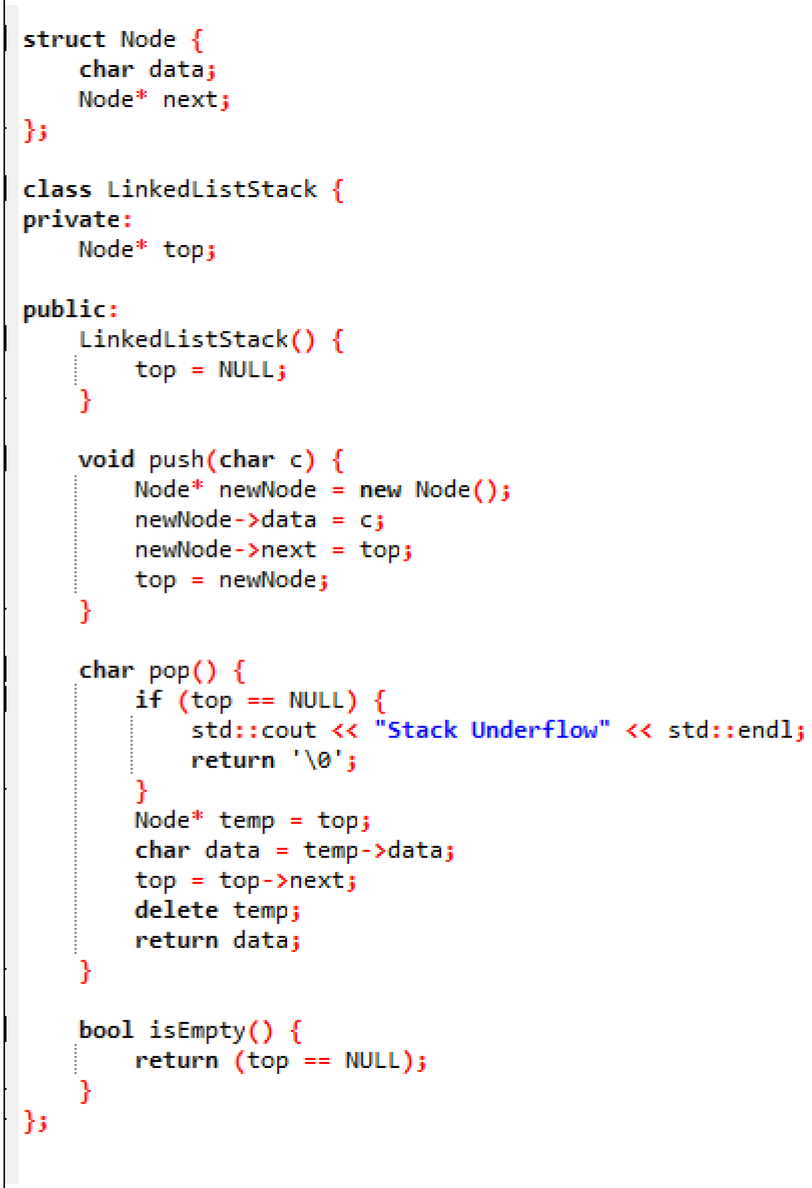
Prelim Skills Exam

Course Code: CPE010	**Program:** Computer Engineering
Course Title: Data Structures and Algorithms	**Date Performed: 9/2/25**
Section: CPE21S4	**Date Submitted: 9/2/25**
Name(s): Quioyo, Angelo M.	**Instructor: Engr. Jimlord Quejado**

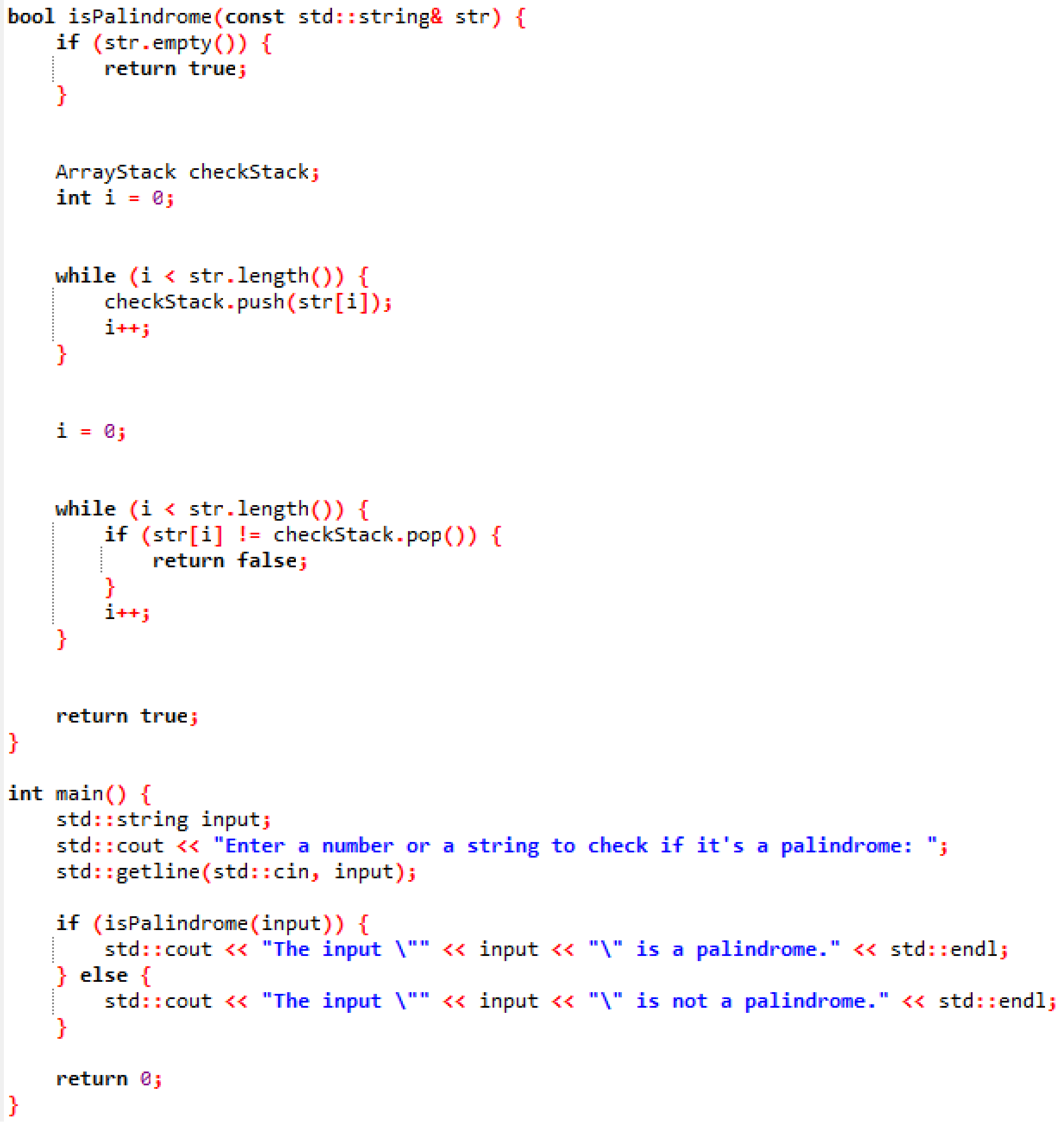
Code:



```
#include <iostream>
#include <string>

#define MAX_SIZE 100

class ArrayStack {
private:
    char arr[MAX_SIZE];
    int top;

public:
    ArrayStack() {
        top = -1;
    }

    void push(char c) {
        if (top >= MAX_SIZE - 1) {
            std::cout << "Stack Overflow" << std::endl;
            return;
        }
        arr[++top] = c;
    }

    char pop() {
        if (top < 0) {
            std::cout << "Stack Underflow" << std::endl;
            return '\0';
        }
        return arr[top--];
    }

    bool isEmpty() {
        return (top < 0);
    }
};


struct Node {
    char data;
    Node* next;
};
```

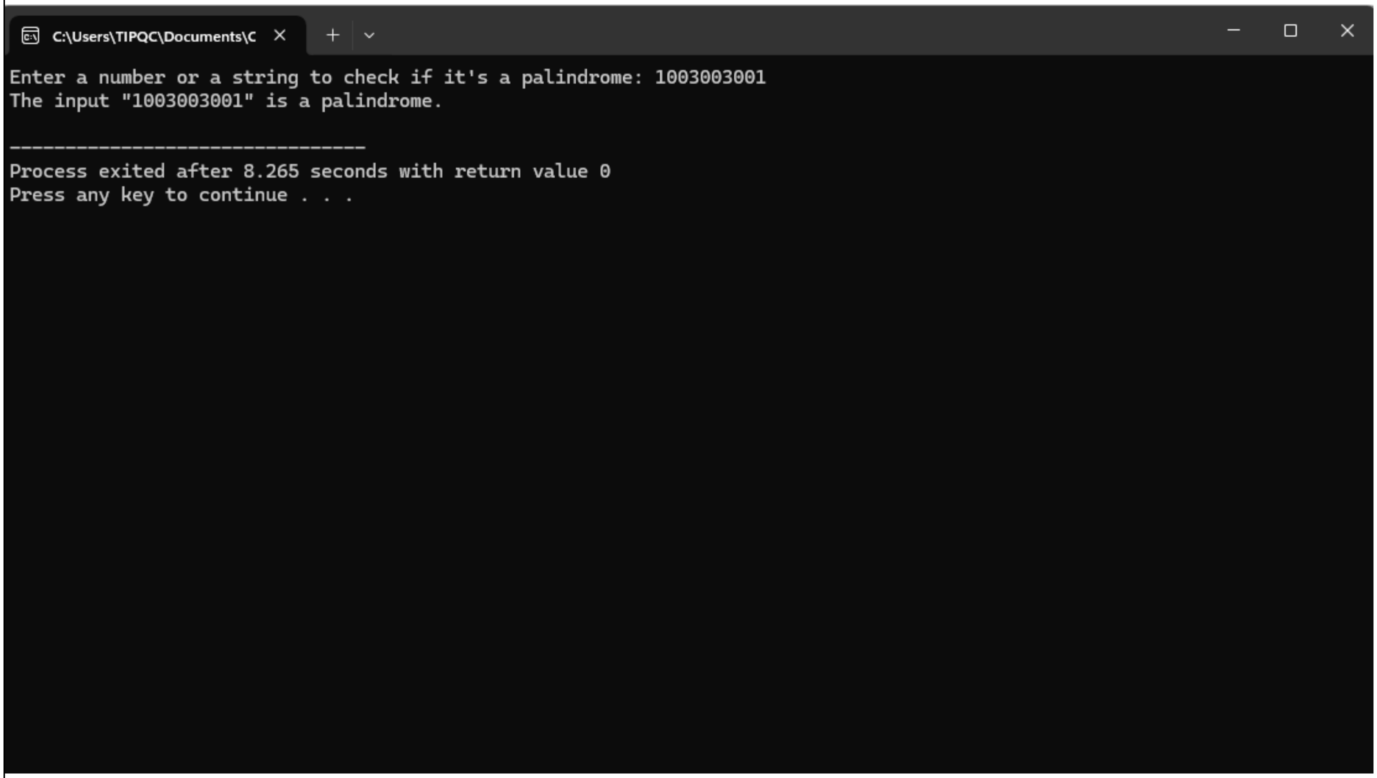


```cpp
struct Node {
    char data;
    Node* next;
};

class LinkedListStack {
private:
    Node* top;

public:
    LinkedListStack() {
        top = NULL;
    }

    void push(char c) {
        Node* newNode = new Node();
        newNode->data = c;
        newNode->next = top;
        top = newNode;
    }

    char pop() {
        if (top == NULL) {
            std::cout << "Stack Underflow" << std::endl;
            return '\0';
        }
        Node* temp = top;
        char data = temp->data;
        top = top->next;
        delete temp;
        return data;
    }

    bool isEmpty() {
        return (top == NULL);
    }
};
```

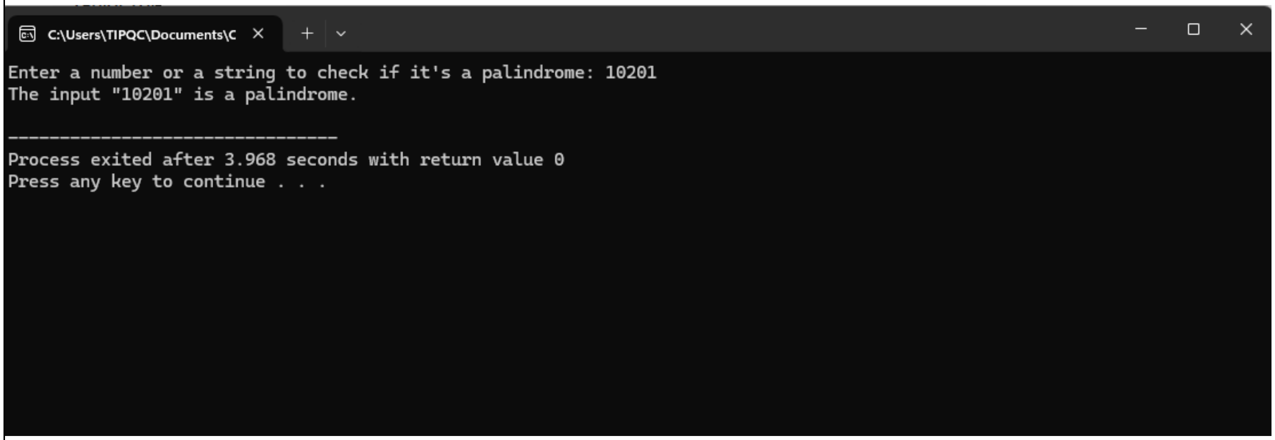


```cpp
bool isPalindrome(const std::string& str) {
    if (str.empty()) {
        return true;
    }


    ArrayStack checkStack;
    int i = 0;


    while (i < str.length()) {
        checkStack.push(str[i]);
        i++;
    }


    i = 0;


    while (i < str.length()) {
        if (str[i] != checkStack.pop()) {
            return false;
        }
        i++;
    }


    return true;
}

int main() {
    std::string input;
    std::cout << "Enter a number or a string to check if it's a palindrome: ";
    std::getline(std::cin, input);

    if (isPalindrome(input)) {
        std::cout << "The input \"" << input << "\" is a palindrome." << std::endl;
    } else {
        std::cout << "The input \"" << input << "\" is not a palindrome." << std::endl;
    }

    return 0;
}
```

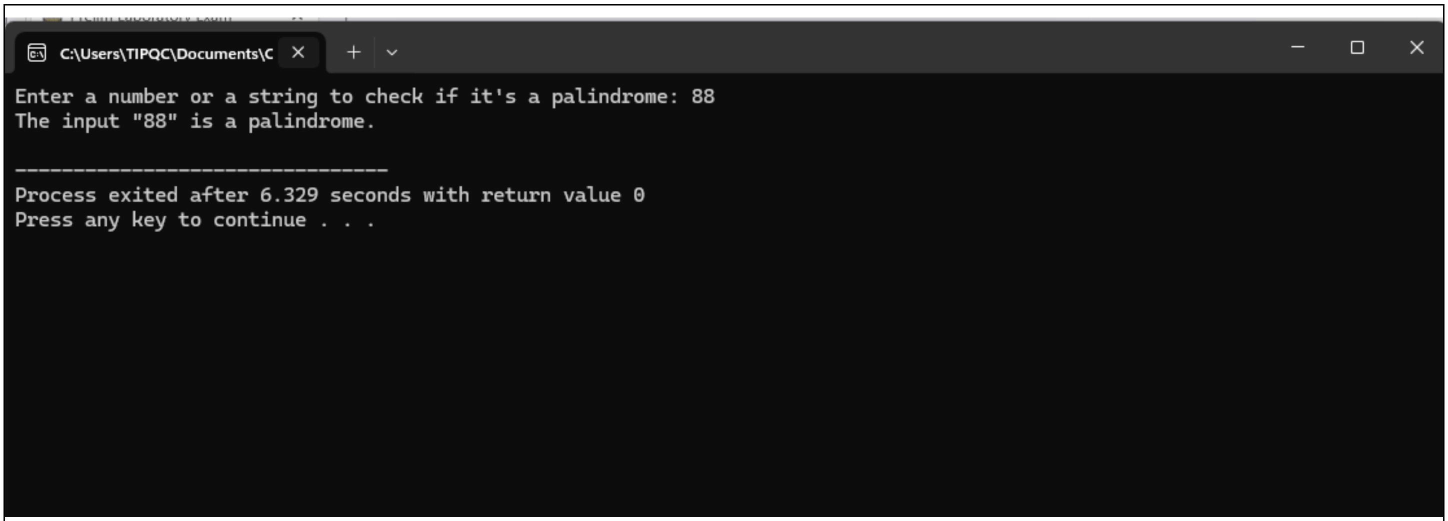
OUTPUT:



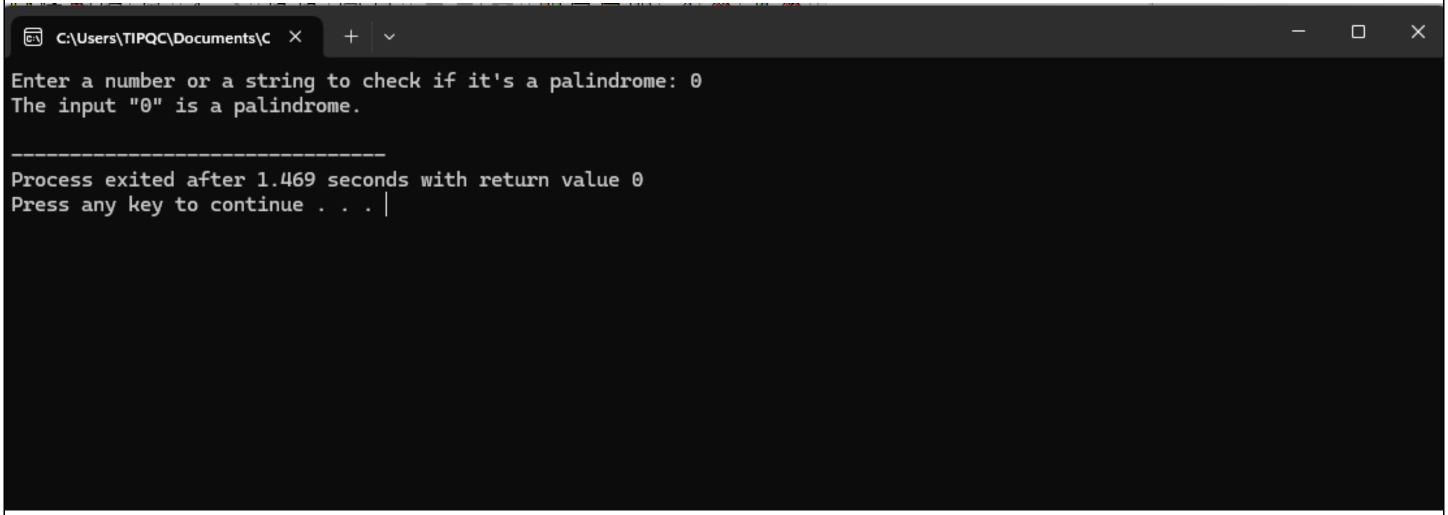
```
Enter a number or a string to check if it's a palindrome: 1003003001
The input "1003003001" is a palindrome.

--------------------------------
Process exited after 8.265 seconds with return value 0
Press any key to continue . . .
```



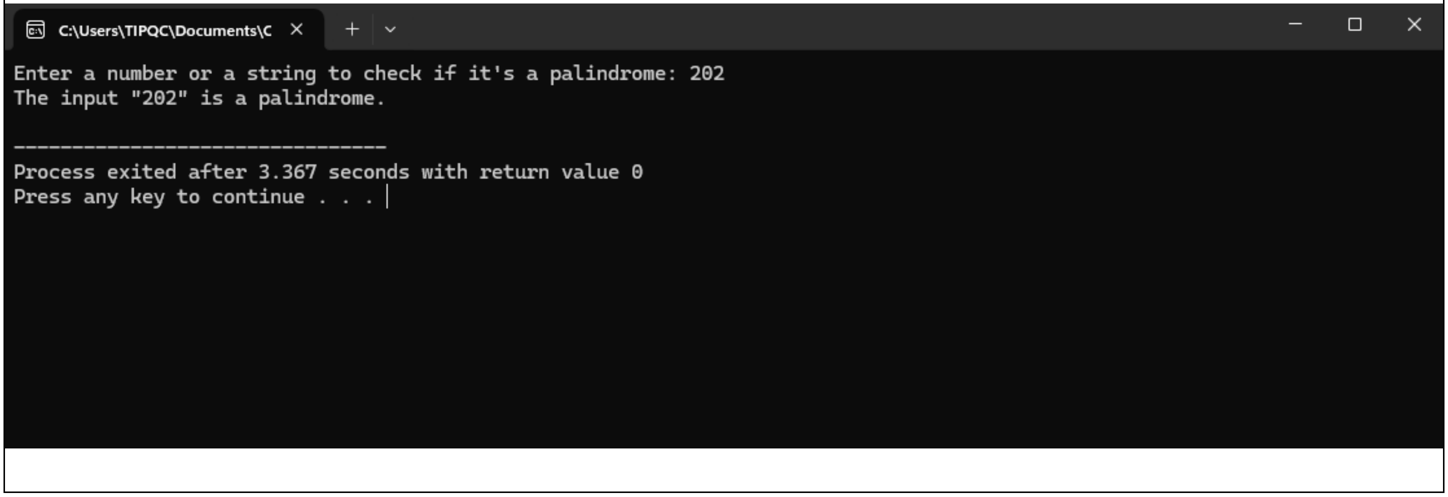
```
Enter a number or a string to check if it's a palindrome: 10201
The input "10201" is a palindrome.

--------------------------------
Process exited after 3.968 seconds with return value 0
Press any key to continue . . .
```



```
Enter a number or a string to check if it's a palindrome: 88
The input "88" is a palindrome.

--------------------------------
Process exited after 6.329 seconds with return value 0
Press any key to continue . . .
```



```
Enter a number or a string to check if it's a palindrome: 0
The input "0" is a palindrome.

--------------------------------
Process exited after 1.469 seconds with return value 0
Press any key to continue . . .
```



```
Enter a number or a string to check if it's a palindrome: 202
The input "202" is a palindrome.

--------------------------------
Process exited after 3.367 seconds with return value 0
Press any key to continue . . .
```

9. Assessment Rubric