NAME : Rehan Ahmed

Sapp : 54606

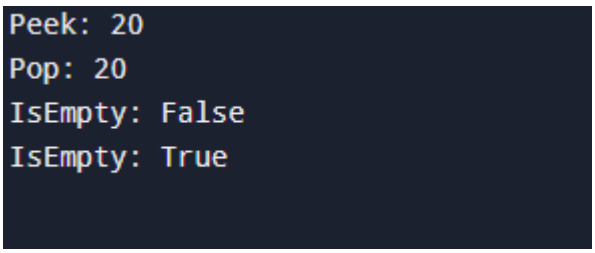
Lab task 04

QUESTION # 01 :

```
#include <iostream>
#include <conio.h>
using namespace std;
#define MAX_SIZE 100

class Stack {
private:
   int data[MAX_SIZE];
   int top;
public:
   Stack(int ignored = 0) {
      top = -1;
   }

   void push(const int element) {
      if (top == MAX_SIZE - 1) {
         throw runtime_error("Error: Stack overflow!");
      }
      data[++top] = element;
   }


   int pop() {
      if (top == -1) {
         throw runtime_error("Error: Stack underflow!");
      }
      return data[top--];
   }

      int peek() {
      if (top == -1) {
         throw runtime_error("Error: Stack is empty!");
      }
      return data[top];
   }
   void clear() {
      top = -1;
   }
```

```
    bool isEmpty() {
        return top == -1;
    }
};

int main() {
    Stack stack;
    try {
        stack.push(10);
        stack.push(20);
        cout << "Peek: " << stack.peek() << endl;
        cout << "Pop: " << stack.pop() << endl;
        cout << "IsEmpty: " << (stack.isEmpty() ? "True" : "False") << endl;
        stack.clear();
        cout << "IsEmpty: " << (stack.isEmpty() ? "True" : "False") << endl;
    } catch (const runtime_error& e) {
        cerr << "Error: " << e.what() << endl;
    }
    return 0;
}
```

OUTPUT



Question no 2

```
#include <iostream>

using namespace std;

#define MAX_SIZE 100


class Stack {

private:

    char data[MAX_SIZE];

    int top;
```

```
public:
    Stack() {
        top = -1;
    }

    void push(char element) {
        if (top == MAX_SIZE - 1) {
            cout << "Error: Stack overflow!" << endl;
            exit(1);
        }
        data[++top] = element;
    }

    char pop() {
        if (top == -1) {
            cout << "Error: Stack underflow!" << endl;
            exit(1);
        }
        return data[top--];
    }
};

void reverse_string(char* str) {
    Stack stack;

    int len = strlen(str);
    for (int i = 0; i < len; i++) {
        stack.push(str[i]);
    }
```

```
    for (int i = 0; i < len; i++) {
        str[i] = stack.pop();
    }
}

int main() {
    char str[MAX_SIZE];
    cout << "Enter a string: ";
    cin.getline(str, MAX_SIZE);

    cout << "Original string: " << str << endl;
    reverse_string(str);
    cout << "Reversed string: " << str << endl;

    return 0;
}
```

Output: