Assignment_DSA_LAB_04

Kashan Shahid

53686

CS 3-1

Question1:

Code:

```
#include <iostream>

using namespace std;


class Stack {

private:

    int *arr;

    int top;

    int capacity;


public:

    // Constructor: Creates an empty stack

    Stack(int size = 100) {

        arr = new int[size];

        top = -1;
```

```
    capacity = size;
}

// Destructor: Deallocates memory used by the stack
~Stack() {
    delete[] arr;
}

// Push: Adds an element to the top of the stack
void push(int data) {
    if (top == capacity - 1) {
        cout << "Stack Overflow\n";
        return;
    }
    arr[++top] = data;
}

// Pop: Removes the element from the top of the stack
void pop() {
    if (isEmpty()) {
```

```
        cout << "Stack Underflow\n";
        return;
    }
    top--;
}

// Peek: Returns the element at the top of the stack
int peek() {
    if (isEmpty()) {
        cout << "Stack is empty\n";
        return -1;
    }
    return arr[top];
}

// Clear: Removes all elements from the stack
void clear() {
    top = -1;
}
```

```
    // isEmpty: Returns true if the stack is empty, otherwise false
    bool isEmpty() {
        return top == -1;
    }
};

int main() {
    Stack stack;

    stack.push(10);
    stack.push(20);
    stack.push(30);

    cout << "Top element: " << stack.peek() << endl;

    stack.pop();
    cout << "Top element after pop: " << stack.peek() << endl;

    stack.clear();
```

```
    cout << "Is stack empty after clear? " << (stack.isEmpty() ? "Yes" : "No") << endl;


    return 0;
}
```

Output:

```
/tmp/tX6qy05aTt.o
Top element: 30
Top element after pop: 20
Is stack empty after clear? Yes


=== Code Execution Successful ===
```

Question_2:

Code:

```
#include <iostream>
#include <stack>
using namespace std;

void reverseString(string &str) {
    stack<char> s;
```

```
    // Push each character of the string into the stack
    for (char ch : str) {
        s.push(ch);
    }

    // Pop each character from the stack and modify the string
    int i = 0;
    while (!s.empty()) {
        str[i++] = s.top();
        s.pop();
    }
}

int main() {
    string str = "Data Structures";

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

```
    return 0;
}
```

Output:

```
/tmp/Do1Oq8Ovld.o
Original string: Data Structures
Reversed string: serutcurtS ataD


=== Code Execution Successful ===
```