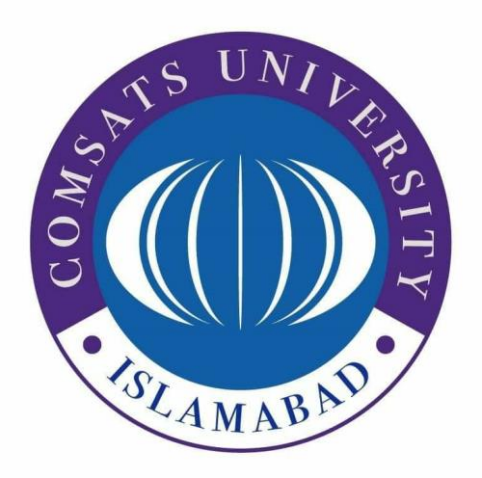


Name: Rao Muhammad Rafay

Reg No Sp22-Bcs-067

Instructor: Mam Yasmeen Jana

Date: 25/10/2023

LAb Mid

COMSATS University Islamabad

Vehari Campus

Question No 1:

To create a single linked list using “SSL” Function …

Answer:

```
#include <iostream>

using namespace std;


const int MAX_SIZE = 5;


class Stack{

private:

    int top;

    int arr[MAX_SIZE];


public:

    Stack() {

        top = -1;

    }

    bool isEmpty() {

        return top == -1;

    }

    bool isFull() {

        return top == MAX_SIZE - 1;

    }
```

```
void push(int data) {
    if (isFull()) {
        cout << "Stack is full. Cannot push." << endl;
        return;
    }
    arr[++top] = data;
}
void pop() {
    if (isEmpty()) {
        cout << "Stack is empty. Cannot pop." << endl;
        return;
    }
    --top;
}
 int peek() {
    if (isEmpty()) {
        cout << "Stack is empty. Cannot pop." << endl;
        return 0;
    }
    //cout<<arr[top]<<"  ";
    return arr[top];
}
int display() {
```

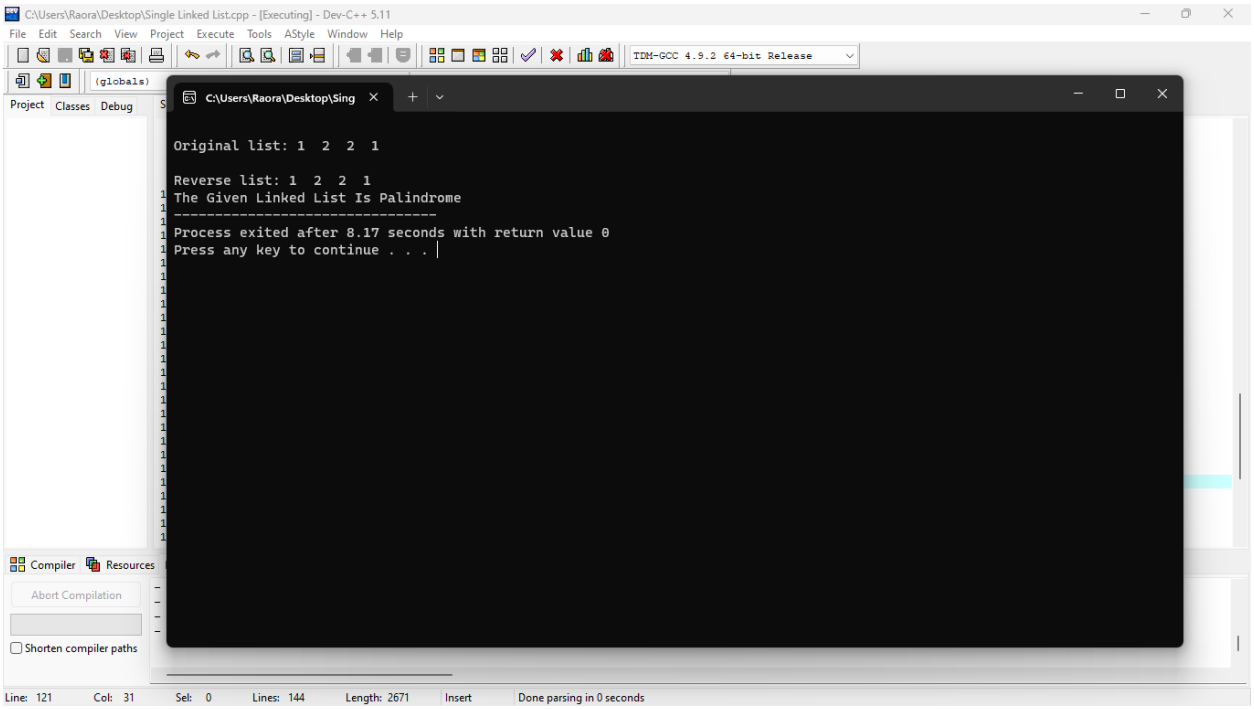
```
        if (isEmpty()) {
            cout << "Stack is empty. Cannot Display." << endl;
            return -1;
        }
        int display=top;
        while(display!=-1){
            cout<<arr[display]<<"  ";
            display--;
                    }
    }
};



class Node {
public:
    int data;
    Node* next;

    Node(int data) {
        this->data = data;
        this->next = NULL;
    }
};
```

```
class LinkedList {
public:
    Node* head;
        Stack stack;
    LinkedList() {
        head = NULL;
    }

    void insert(int data) {
        Node* newNode = new Node(data);
        if (head == NULL) {
            head = newNode;
        }
        else {
            Node* temp = head;
            while (temp->next != NULL) {
                temp = temp->next;
            }
            temp->next = newNode;
        }
    }
        void SSL(){
```

```
        Node* temp=head;
        while(temp!=NULL){
                stack.push(temp->data);
                temp=temp->next;
        }

        stack.display();
}
void palindrome(){
        Node* temp=head;
        int a=0;
        while(temp!=NULL){
        if(temp->data==stack.peek()){
                temp=temp->next;
                stack.pop();
        }
        else{
                cout<<"\nThe Given Linked List Is Not Palindrome"<<endl;
                a=5;
                return;
        }
        }
        if(a==0){
```

```
                cout<<"\nThe Given Linked List Is Palindrome";
            }

        }


    void display() {
        Node* temp = head;
        while (temp != NULL) {
            cout << temp->data << "  ";
            temp = temp->next;
        }
        cout<<endl;
    }

};

int main() {
    LinkedList list;
    list.insert(1);
    list.insert(2);
    list.insert(2);
    list.insert(1);
```

```
    cout << "\nOriginal list: ";
    list.display();
    cout << "\nReverse list: ";
    list.SSL();
        list.palindrome();
    return 0;
}
```



Question no 2:

Simple Stack function

Answer:

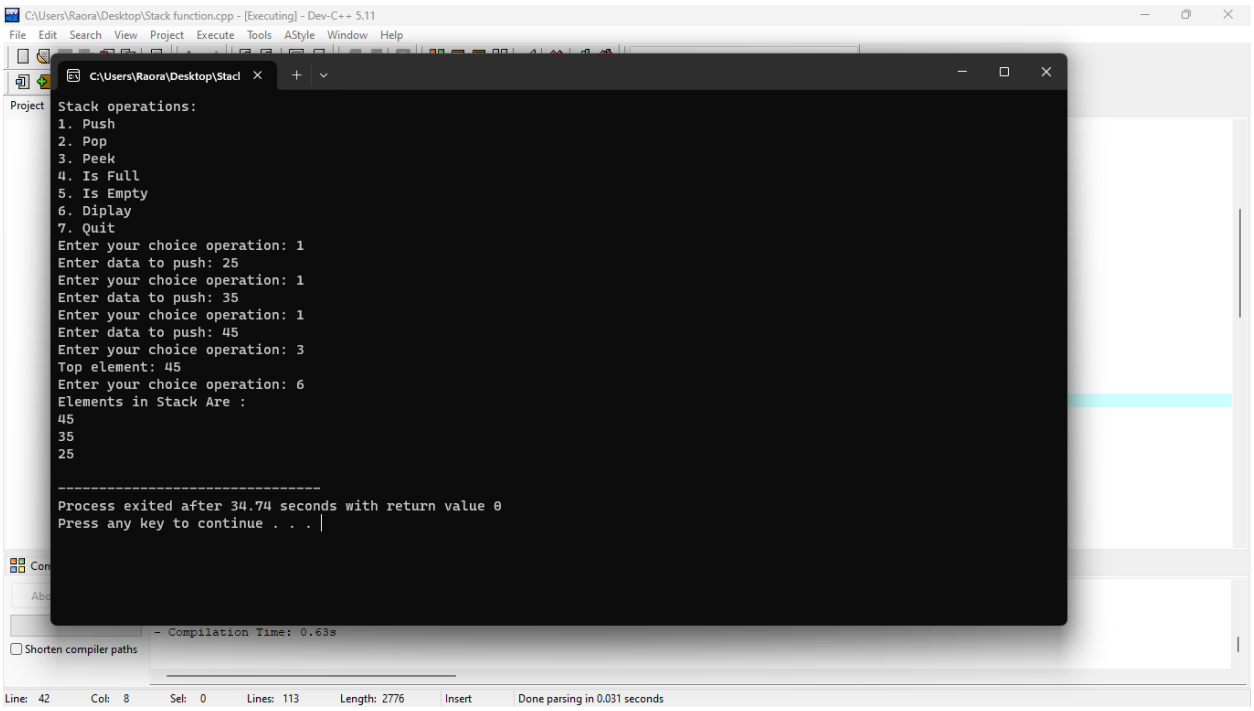
```
#include <iostream>
using namespace std;

const int MAX_SIZE = 100;

class Stack{
private:
    int top;
    int arr[MAX_SIZE];

public:
    Stack() {
        top = -1;
    }
    bool isEmpty() {
        return top == -1;
    }
    bool isFull() {
        return top == MAX_SIZE - 1;
    }
    void push(int data) {
        if (isFull()) {
            cout << "Stack is full. Cannot push." << endl;
```

```
        return;
    }
    arr[++top] = data;
}
void pop() {
    if (isEmpty()) {
        cout << "Stack is empty. Cannot pop." << endl;
        return;
    }
    --top;
}
int peek() {
    if (isEmpty()) {
        cout << "Stack is empty. Cannot peek." << endl;
        return -1;
    }
    return arr[top];
}
int display() {
    if (isEmpty()) {
        cout << "Stack is empty. Cannot Display." << endl;
        return -1;
    }
```

```
        int display=top;
        while(display!=-1){
            cout<<arr[display]<<endl;
            display--;
                    }
    }
};

int main() {
    Stack stack;

    cout << "Stack operations:" << endl;
    cout << "1. Push" << endl;
    cout << "2. Pop" << endl;
    cout << "3. Peek" << endl;
    cout << "4. Is Full" << endl;
    cout << "5. Is Empty" << endl;
    cout << "6. Diplay" << endl;
    cout << "7. Quit" << endl;

    int choice, data;
```

```
do {
    cout << "Enter your choice operation: ";
    cin >> choice;

    switch (choice) {
        case 1:
            cout << "Enter data to push: ";
            cin >> data;
            stack.push(data);
            break;
        case 2:
            stack.pop();
            break;
        case 3:
            cout << "Top element: " << stack.peek() << endl;
            break;
        case 4:
            if (stack.isFull()) {
                cout << "Stack is full." << endl;
            } else {
                cout << "Stack is not full." << endl;
            }
            break;
```

```
            case 5:
                if (stack.isEmpty()) {
                    cout << "Stack is empty." << endl;
                } else {
                    cout << "Stack is not empty." << endl;
                }
                break;
            case 7:
                cout << "Exiting program." << endl;
                break;
            case 6:
                cout << "Elements in Stack Are :" << endl;
                stack.display();
                break;
            default:
                cout << "Invalid choice. Please try again." << endl;
        }
    } while (choice != 6);

    return 0;
}
```


C:\Users\Raora\Desktop\Stack function.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
Project
C:\Users\Raora\Desktop\Stacl
Stack operations:
1. Push
2. Pop
3. Peek
4. Is Full
5. Is Empty
6. Diplay
7. Quit
Enter your choice operation: 1
Enter data to push: 25
Enter your choice operation: 1
Enter data to push: 35
Enter your choice operation: 1
Enter data to push: 45
Enter your choice operation: 3
Top element: 45
Enter your choice operation: 6
Elements in Stack Are :
45
35
25
Process exited after 34.74 seconds with return value 0
Press any key to continue . . .
- Compilation Time: 0.63s
Shorten compiler paths
Line: 42 Col: 8 Sel: 0 Lines: 113 Length: 2776 Insert Done parsing in 0.031 seconds