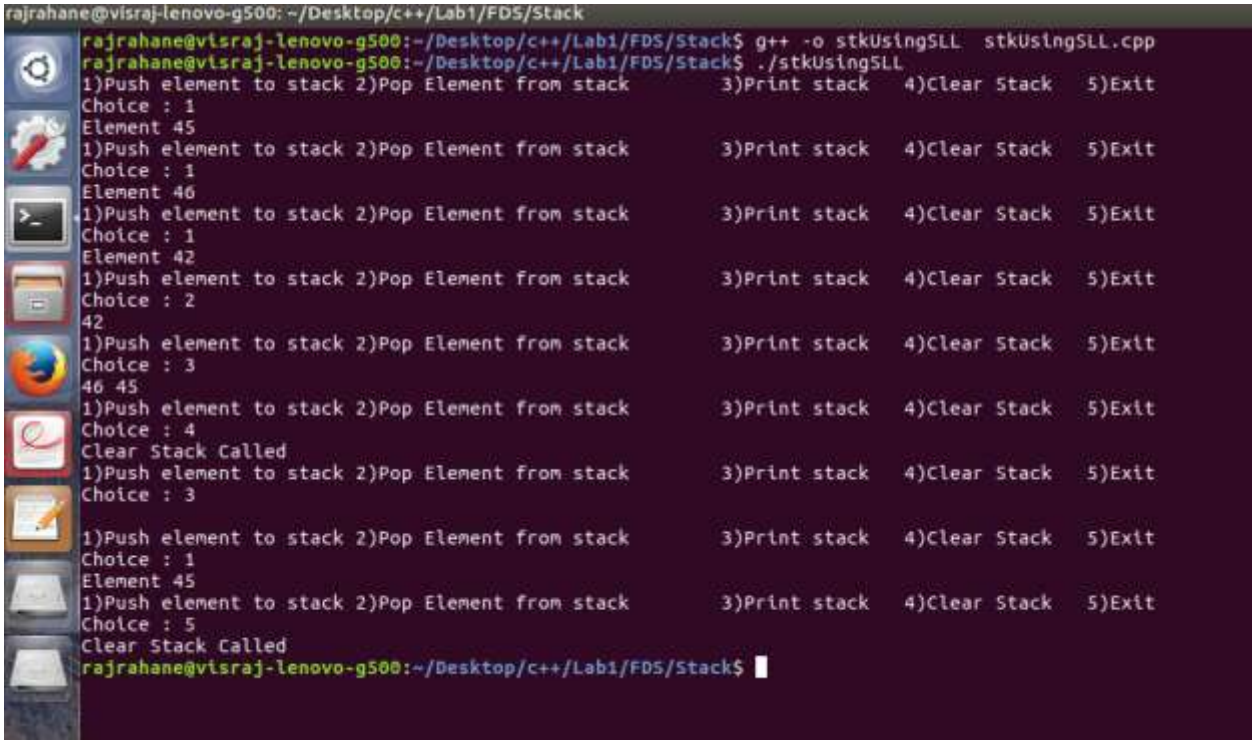
Rajvaibhav Rahane
17u283 223045
SE-C Comp,Viit,Pune

CODE:

```
/*
*
*@Rajvaibhav Rahane
*/
/*
*	Program to Demonstrate Stack Operations using SLL
*	implements push,pop,clearStack,printStack methods.
*/
#include<iostream>
using namespace std;
struct Node{
	int data;
	Node *next;
};
int popElementFromStack(Node **head){		//pop element from stack, pass by ref
	if(*head!=NULL){
		Node *removedNode=*head;
		*head=(*head)->next;
		int element=removedNode->data;
		delete removedNode;
		return element;
	}else{
		return -1;
	}
}
void pushElementToStack(Node **head,int element){ //push to stack, pass by ref
	Node *newNode=new Node;
	newNode->next=*head;
	newNode->data=element;
	*head=newNode;
}
void printStack(Node *head){			//print stack, pass by value
	Node *temp=head;
	while(temp!=NULL){
		cout<<temp->data<<" ";
		temp=temp->next;
	}
	cout<<endl;
}
void clearStack(Node **head){			//clear Stack, pass by ref
	Node *temp;
	while(*head!=NULL){
```

```
                temp=*head;
                *head=(*head)->next;
                delete temp;
        }
        cout<<"Clear Stack Called\n";
}
void printMenu(){                       //print menu
        cout<<"1)Push element to stack\t";
        cout<<"2)Pop Element from stack\t";
        cout<<"3)Print stack\t";
        cout<<"4)Clear Stack\t";
        cout<<"5)Exit\n";
}
int main(){                             //ui
        Node *head=NULL;
        int choice,element;
        do{
                printMenu();
                cout<<"Choice : ";cin>>choice;
                switch(choice){
                        case 1:{
                                cout<<"Element ";cin>>element;
                                pushElementToStack(&head,element);
                                break;
                        }
                        case 2:{
                                element=popElementFromStack(&head);
                                if(element!=-1)cout<<element<<endl;
                                else cout<<"Empty Stack\n";
                                break;
                        }
                        case 3:{
                                printStack(head);
                                break;
                        }
                        case 4: case 5:{
                                clearStack(&head);
                                break;
                        }
                        default:cout<<"Invalid Input,Try again\n";
                }
        }while(choice!=5);
        return 0;
}
```

Output:


rajrahane@visraj-lenovo-g500: ~/Desktop/c++/Lab1/FDS/Stack
rajrahane@visraj-lenovo-g500:~/Desktop/c++/Lab1/FDS/Stack$ g++ -o stkUsingSLL stkUsingSLL.cpp
rajrahane@visraj-lenovo-g500:~/Desktop/c++/Lab1/FDS/Stack$ ./stkUsingSLL
1)Push element to stack 2)Pop Element from stack 3)Print stack 4)Clear Stack 5)Exit
Choice : 1
Element 45
1)Push element to stack 2)Pop Element from stack 3)Print stack 4)Clear Stack 5)Exit
Choice : 1
Element 46
1)Push element to stack 2)Pop Element from stack 3)Print stack 4)Clear Stack 5)Exit
Choice : 1
Element 42
1)Push element to stack 2)Pop Element from stack 3)Print stack 4)Clear Stack 5)Exit
Choice : 2
42
1)Push element to stack 2)Pop Element from stack 3)Print stack 4)Clear Stack 5)Exit
Choice : 3
46 45
1)Push element to stack 2)Pop Element from stack 3)Print stack 4)Clear Stack 5)Exit
Choice : 4
Clear Stack Called
1)Push element to stack 2)Pop Element from stack 3)Print stack 4)Clear Stack 5)Exit
Choice : 3
1)Push element to stack 2)Pop Element from stack 3)Print stack 4)Clear Stack 5)Exit
Choice : 1
Element 45
1)Push element to stack 2)Pop Element from stack 3)Print stack 4)Clear Stack 5)Exit
Choice : 5
Clear Stack Called
rajrahane@visraj-lenovo-g500:~/Desktop/c++/Lab1/FDS/Stack$