

LAB 02

M. Huzaifa Mustafa

SP22-BSCS-0046

1. Implement Stack.

Source Code:

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

```
int Size_of_Stack;
```

```
class Stack
```

```
{
```

```
    private:
```

```
        int top;
```

```
        int array[];
```

```
    public:
```

```
        Stack(int temp_number)
```

```
        {
```

```
            Size_of_Stack = temp_number;
```

```
            array[temp_number];
```

```
            top = -1;
```

```
        }
```

```
        void push(int a)
```

```
        {
```

```
            if(!is_full())
```

```
            {
```

```
                array[++top] = a;
```

```
            }
```

```
            else
```

```
            {
```

```
                cout<<"Error !!! Stack is full : "<<endl;
```

```
            }
```

```
        }
```

```
        void pop()
```

```
        {
```

```
            if(!is_empty()){
```

```

        cout << "\n" << array[top--] << "\n"<< endl;
    }
    else{
        cout << "ERROR!! Stack is empty " << endl;
    }
}
void peek()
{
    if(!is_empty()){
        cout << "\n" << array[top] << "\n"<< endl;
    }

    else{
        cout << "ERROR!! Stack is empty " << endl;
    }
}
bool is_empty()
{
    if(top != -1){
        return false;
    }
    else{
        return true;
    }
}
bool is_full(){
    if(top < Size_of_Stack){
        return false;
    }

    else{
        return true;
    }
}

};

int main()
{
    int num;
    int option = 1;

```

```
cout<<"Enter the size of the Stack : "<<endl;
```

```
cin>>option;
```

```
Stack arr(num);
```

```
while(option != 0){
```

```
    cout << "\n\n1)Exit \n2)PUSH \n3)POP \n4)PEEK \n\n" << endl;
```

```
    cin >> option;
```

```
    if(option == 1){
```

```
        break;
```

```
    }
```

```
    if(option == 2){
```

```
        int number;
```

```
        cout << "Enter a number: " << endl;
```

```
        cin >> number;
```

```
        arr.push(number);
```

```
    }
```

```
    if(option == 3){
```

```
        arr.pop();
```

```
    }
```

```
    if(option == 4){
```

```
        arr.peak();
```

```
    }
```

```
    }
```

```
}
```

Picture:

```
C:\Users\Administrator\Desktop\Untitled1.exe
Enter the size of the Stack :
5
1)Exit
2)PUSH
3)POP
4)PEEK
2
Enter a number:
1
1)Exit
2)PUSH
3)POP
4)PEEK
3
1
1)Exit
2)PUSH
3)POP
4)PEEK
4
ERROR!! Stack is empty
1)Exit
2)PUSH
3)POP
4)PEEK
1
```

2. Print Reverse String using Stack.

Source Code:

```
#include <iostream>
```

```
using namespace std;
```

```
const int size_of_stack = 100;
```

```
class stack{
```

```
private:
```

```
    int size;
```

```
    int top;
```

```
    char stack1[size_of_stack];
```

```
public:
```

```
    stack(){
```

```
        top=-1;
```

```
    }
```

```
    void push (char a){
```

```
        if(top >= size_of_stack - 1){
```

```
            cout << "Stack is full\n";
```

```
        }
```

```
        else{
```

```
            top++;
```

```
            stack1[top]=a;
```

```
        }
```

```
    }
```

```

        char peek() {
    if (top < 0) {
        cout << "Stack is empty\n";
        return 0;
    }

        else {
            return stack1[top];
        }
    }

    void pop() {
    if (top < 0) {
        cout << "Stack is empty\n";
    }

        else {
            top--;
        }
    }

        bool isFull1(){
    return top == size - 1;
    }

        bool isEmpty1(){
    return top == -1;
    }

};

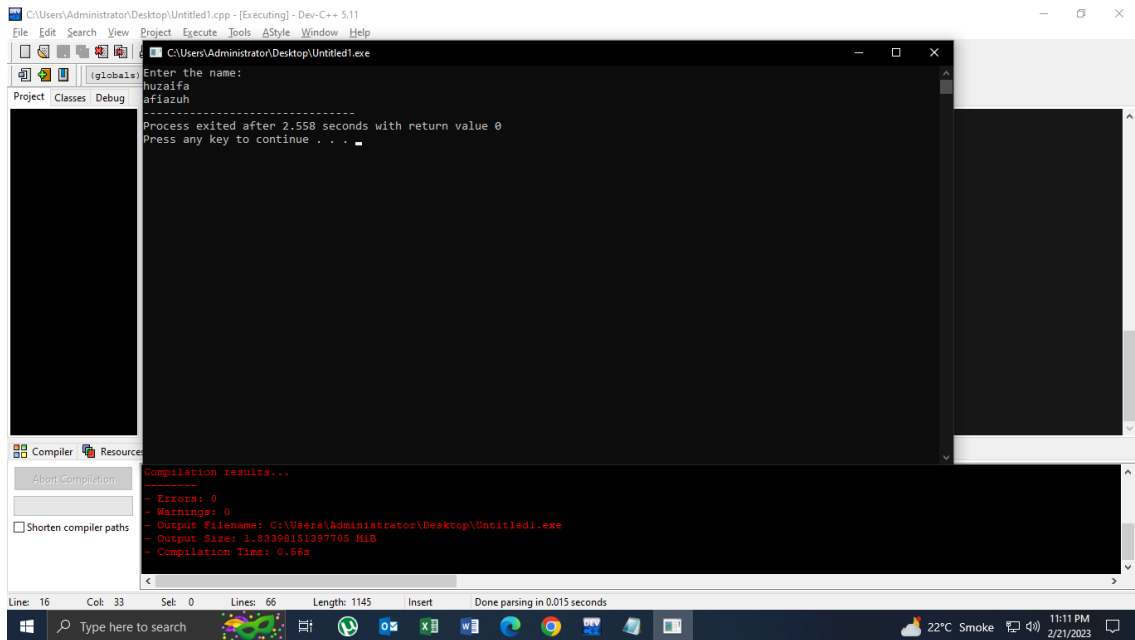
void ReverseString(string name){
    stack s;
    int n= name.length();
    for(int i=0; i<n; i++){
        s.push(name[i]);
    }
    for(int i=0; i<n; i++){
        cout<<s.peek();
        s.pop();
    }
}

int main(){
    string name;
    cout<<"Enter the name: "<<endl;
    cin>>name;
    ReverseString(name);
}

```

}

Picture:



3. Implement 2 Stacks in Single Array.

Source Code:

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

```
int length1, length2;
```

```
class Stack{
```

```
private:
```

```
    int top1, top2 , len ;
    int arr[];
```

```
public:
```

```
    Stack(int l1, int l2){
        arr[l1 + l2];
        len = l1+l2;
        top1 = -1;
        top2 = l1 - 1;
    }
```

```
    bool Full1(){
        return (top1 < length1 - 1);
    }
```

```

bool Full2(){
    return (top2 < len - 1);
}

void Peek1(){
    if( !Empty1() ){
        cout<<"Value at Index " << top1 <<": " <<arr[top1] <<endl;
    }

    else{
        cout<<"\n Stack is Empty." <<endl;
    }
}

void Peek2(){
    if( !Empty2() ){
        cout<<"Value at Index " << top2 <<": " <<arr[top2] <<endl;
    }

    else{
        cout<<"\n Stack is Empty." <<endl;
    }
}

void Push1(int a){

    if( Full1() ){

        arr[++top1] = a;
        cout<<"\n Value at Index " << top1 <<": " <<arr[top1] <<endl;
    }

    else{
        cout<<"\n Stack is Full." <<endl;
    }
}

void Push2(int a){

    if( Full2() ){

```

```

        arr[++top2] = a;
        cout<<"\n Value at Index " << top2 <<": " <<arr[top2] <<endl;
    }

    else{
        cout<<"\n Stack is Full." <<endl;
    }
}

bool Empty1(){
    return (top1 == -1);
}

bool Empty2(){
    return (top2 == length1 - 1);
}

void Pop1(){

    if( !Empty1() ){

        cout<<"\n Value at index "<<top1--<<" : " << arr[top1]<<endl;
    }

    else{
        cout<<"\n Stack is Empty." <<endl;
    }

}

void Pop2(){

    if( !Empty2() ){

        cout<<"\n Value at index "<<top2--<<": " << arr[top2]<<endl;
    }

    else{
        cout<<"\n Stack is Empty." <<endl;
    }

}

```



```
};
```

```
int main(){
```

```
    cout<<"Enter size of Stack 1: ";  
    cin>>length1;
```

```
    cout<<"Enter size of Stack 2: ";  
    cin>>length2;
```

```
    Stack s1(length1, length2);
```

```
    int choice = 0, option = 0;
```

```
    while(choice != 3){
```

```
        cout<<"\n 01. Stack 01 \n 02. Stack 02 \n 03. Exit \n ";  
        cin>>choice;
```

```
        if(choice == 1){
```

```
            cout<<"\n 01. Push \n 02. Pop \n 03.Peek \n 04.Exit \n ";  
            cin>>option;
```

```
            switch(option){
```

```
                case 1:{
```

```
                    int val;
```

```
                    cout<<"\n Enter value: ";  
                    cin>>val;
```

```
                    s1.Push1(val);  
                    break;
```

```
                }
```

```
                case 2:{
```

```
                    s1.Pop1();  
                    break;
```

```
                }
```

```
                case 3:{
```

```
                    s1.Peek1();
```

```

        break;
    }

    case 4:{
        break;
    }
}

else{

cout<<"\n 01. Push \n 02. Pop \n 03. Peek \n 04. Exit \n ";
cin>>option;

switch(option){
    case 1:{
        int val2;

        cout<<"Enter value : ";
        cin>>val2;

        s1.Push2(val2);
        break;
    }

    case 2:{

        s1.Pop2();
        break;
    }

    case 3:{
        s1.Peek2();
        break;
    }

    case 4:{
        break;
    }
}

}

}

}

```

Picture:

```

Select C:\Users\Administrator\Desktop\Untitled2.exe
Enter size of Stack 1: 3
Enter size of Stack 2: 2

01. Stack 01
02. Stack 02
03. Exit
1

01. Push
02. Pop
03. Peek
04. Exit
1

Enter value: 1

Value at Index 0: 1

01. Stack 01
02. Stack 02
03. Exit
2

01. Push
02. Pop
03. Peek
04. Exit
2

Stack is Empty.

01. Stack 01
02. Stack 02
03. Exit
3

01. Push
02. Pop
03. Peek
04. Exit
4

```