

Modify the stack class given in the previous lab to add the exception when user tries to add item while the stack is full and when user tries to add item while the stack is empty. Throw exception in both of the cases and handle these exceptions.

```
#include<iostream>
#include<cstring>
#define SUCCESS 0;
using namespace std;
#define SIZE 2
class exc{
public:
    string info;
};
template<class T>
class Stack
{
private:
    T st[SIZE];
    int top;
public:
    Stack();
    void push(T var);
    T pop();
};

template<class T>
Stack<T>::Stack()
{
    top=-1;
}

template<class T>
void Stack<T>::push(T var)
{
    try
    {
        if(top >= (SIZE-1))
        {
            top = SIZE - 1;
            exc e;
```

```

        e.info="Stack is full";
        throw e;
    }
    else
    {
        st[++top]=var;
    }
}
catch(exc e)
{
    cerr << e.info << endl;
}
}

```

```

template<class t>
t Stack<t>::pop()
{
    try {
        if (top < 0)
        {
            exc e;
            e.info ="stack is empty";
            throw e;
        }
        else
        {
            return st[top--];
        }
    }
    catch(exc e)
    {
        cerr << e.info << endl;
    }
}

```

```

int main()
{
    Stack<float> s1;

    s1.push(111.1F);

```

```
s1.push(222.2F);  
s1.push(333.3F);
```

```
cout<<"1 : "<<s1.pop()<<endl;  
cout<<"2 : "<<s1.pop()<<endl;  
cout<<"3 : "<<s1.pop()<<endl;
```

```
Stack<long> s2;
```

```
s2.push(123123123L);  
s2.push(234234234L);  
s2.push(345345345L);
```

```
cout<<"1 : "<<s2.pop()<<endl;  
cout<<"2 : "<<s2.pop()<<endl;  
cout<<"3 : "<<s2.pop()<<endl;
```

```
return SUCCESS;  
}
```