```cpp
#include <iostream>

using namespace std;
class Stack
{
private :
    int stackSize ;
    int stackTop;
    int *stackArray;

public :
    Stack()
    {
        stackSize=6;
        stackTop=-1;
        stackArray=new int [stackSize] {0};
    }
    Stack(int UserStackSize)
    {
        stackSize=UserStackSize;
        stackTop=-1;
        stackArray=new int [stackSize] {0};
    }
    Stack(Stack& copiedStack)
    {
        stackSize=copiedStack.stackSize;
        stackTop=copiedStack.stackTop;
        stackArray=new int [stackSize] {0};
        for (int i=0;i<stackSize;i++){
            stackArray[i]=copiedStack.stackArray[i];
```

main.cpp

Run

```cpp
        };
    }
    int getTop(){
        return stackTop;
    }
    int pushIntoStack (int data ){
        if (stackTop<(stackSize-1)){
            stackTop++;
            stackArray[stackTop]=data;

            cout<<"the element you pushed is "<<stackArray[stackTop]<<endl;

        }
        else {
            cout<<"Sorry..the stack is already full you can't push into it "<<endl;
        }
    }
    int popFromStack(){
        if (stackTop!=-1){
            int element=stackArray[stackTop];
            stackTop--;
            cout<<"the element you poped is "<<element<<endl;
            return element;

        }
        else {
            cout<<"Sorry..the stack is already empty you can't pop from if "<<endl;
        }
    }
    void printFromStack(){
```

main.cpp

Run

```
60      void printFromStack(){
61          for(int i=0; i <=stackTop; i++){
62              cout<<"the element you pushed in index "<<i<<"is"<<stackArray[i]<<endl;
63          }
64      }
65      ~Stack (){
66          cout<<"END"<<endl;
67          delete [] stackArray;
68      }
69
70  }; //end stack class :)
71  //void popTesting(Stack testedStack);
72  void popTesting(Stack& testedStack);
73  int main()
74  {
75      Stack stack1;
76      stack1.popFromStack();
77      stack1.pushIntoStack(1);
78      stack1.pushIntoStack(5);
79      stack1.pushIntoStack(6);
80      //stack1.printFromStack();
81      stack1.popFromStack();//6
82      stack1.popFromStack();//5
83      stack1.pushIntoStack(7);
84      stack1.popFromStack();//1
85      stack1.popFromStack();//7
86      stack1.popFromStack();//empty
87      stack1.pushIntoStack(7);
88      stack1.popFromStack();//7
89
```

main.cpp

Run

```
88      stack1.popFromStack();//7
89
90      popTesting(stack1);//1
91      stack1.pushIntoStack(6);
92      stack1.pushIntoStack(7);
93
94  //    Stack originalStack=stack1;
95      //stack1.pushIntoStack(20);
96     // stack2.popFromStack();
97      return 0;
98  }
99
100     /*void popTesting(Stack testedStack){
101         testedStack.popFromStack();
102     }*/
103     void popTesting(Stack& testedStack){
104         testedStack.popFromStack();
105     }
106 //pop empty
107 //push 1
108 //push 5
109 //push 6
110 //pop 1
111 //pop 5
112 //push 7
113 //pop 6
114 //pop7
115 //pop empty
```

Output

```
/tmp/NY8l87uswa.o
Sorry..the stack is already empty you can't pop from if
the element you pushed is 1
the element you pushed is 5
the element you pushed is 6
the element you poped is 6
the element you poped is 5
the element you pushed is 7
the element you poped is 7
the element you poped is 1
Sorry..the stack is already empty you can't pop from if
the element you pushed is 7
the element you poped is 7
Sorry..the stack is already empty you can't pop from if
the element you pushed is 6
the element you pushed is 7
END
```