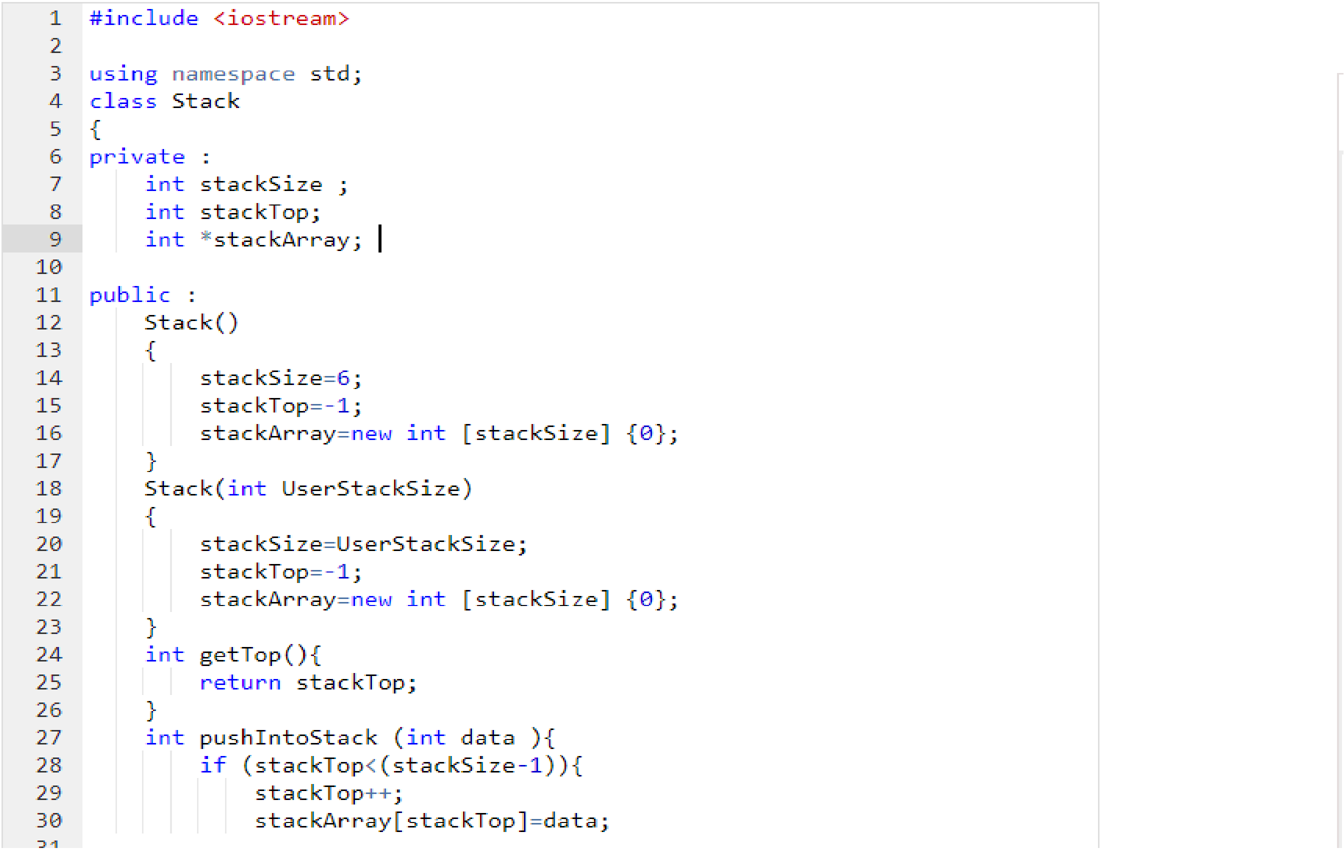
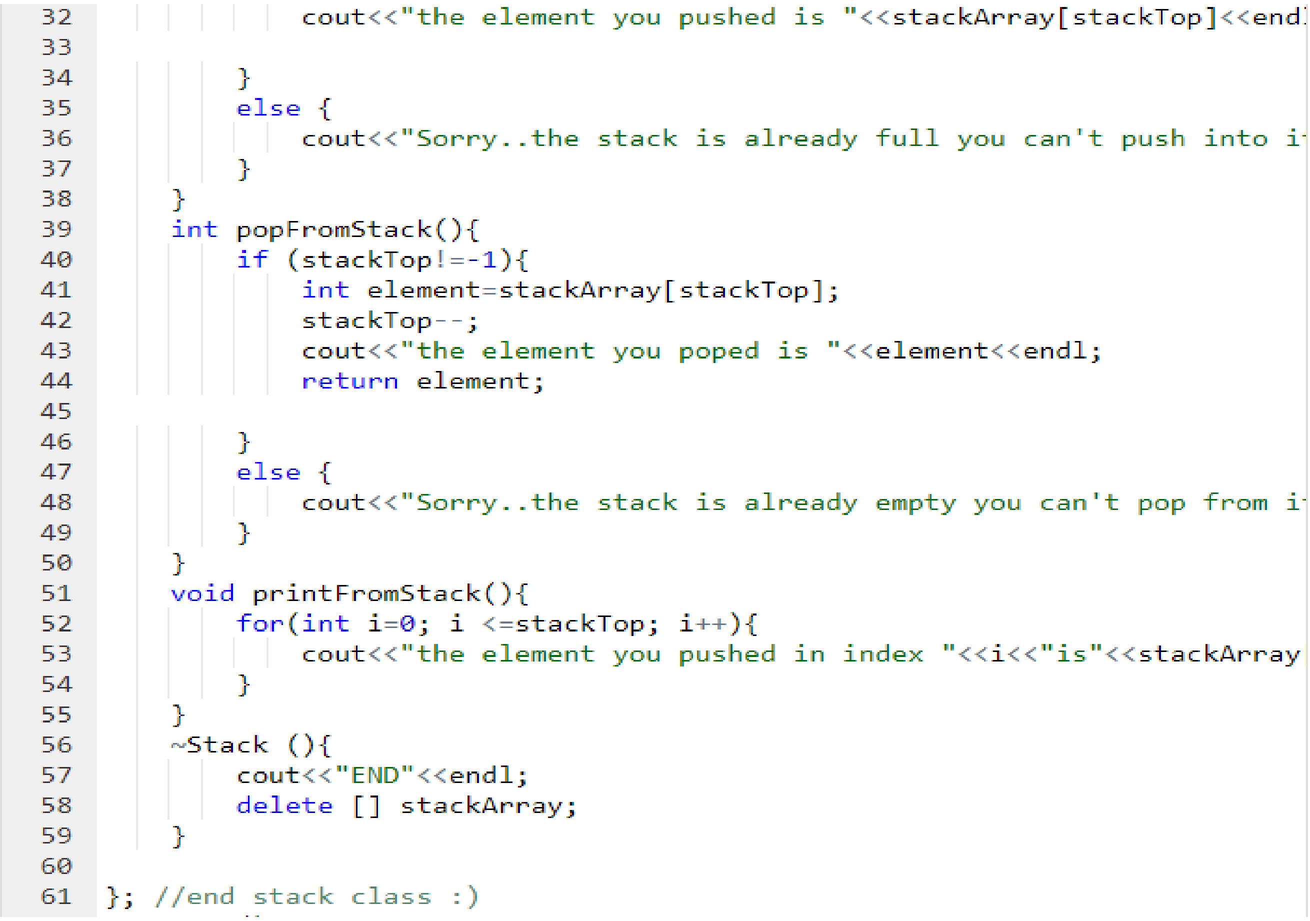
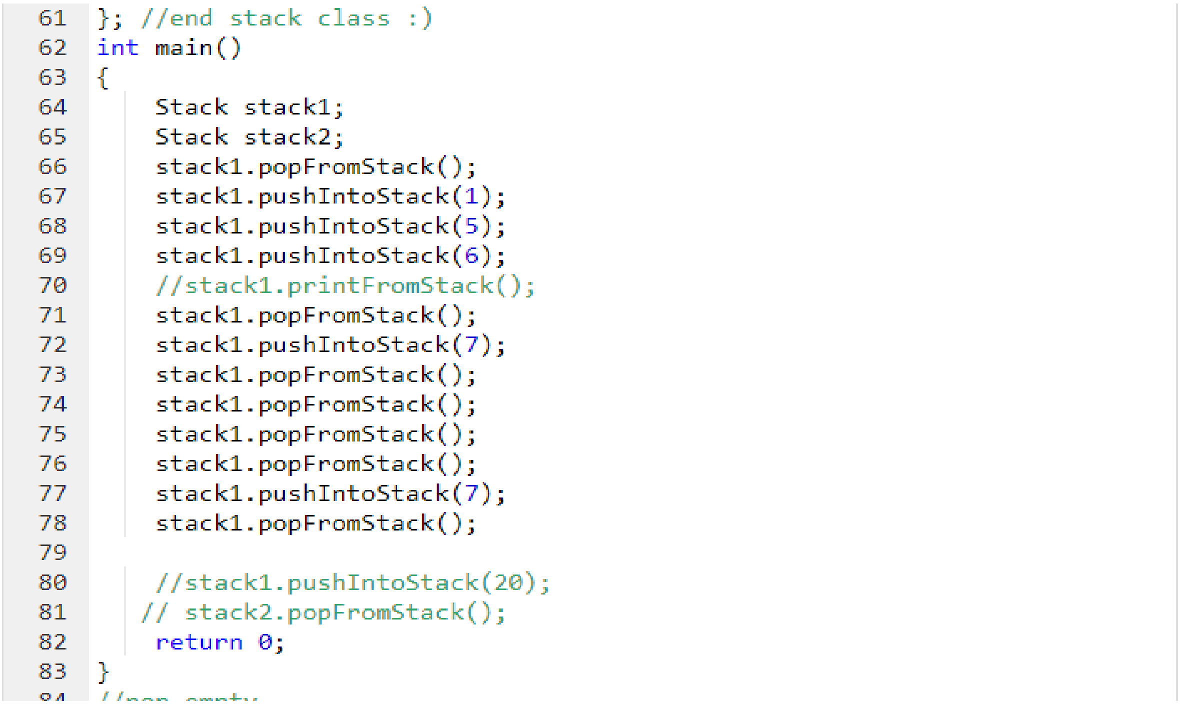


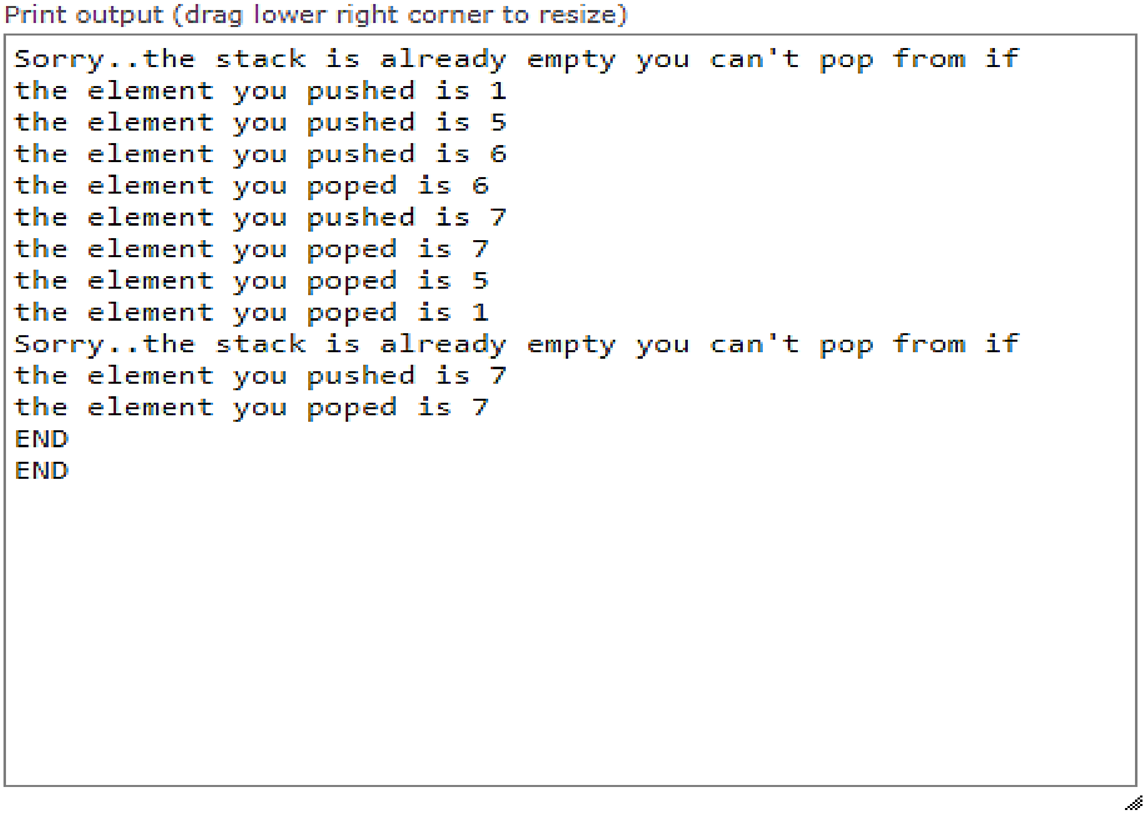
```
#include <iostream>

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



```
            cout<<"the element you pushed is "<<stackArray[stackTop]<<end

        }
        else {
            cout<<"Sorry..the stack is already full you can't push into i
        }
    }
    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 i
        }
    }
    void printFromStack(){
        for(int i=0; i <=stackTop; i++){
            cout<<"the element you pushed in index "<<i<<"is"<<stackArray
        }
    }
    ~Stack (){
        cout<<"END"<<endl;
        delete [] stackArray;
    }

}; //end stack class :)
```

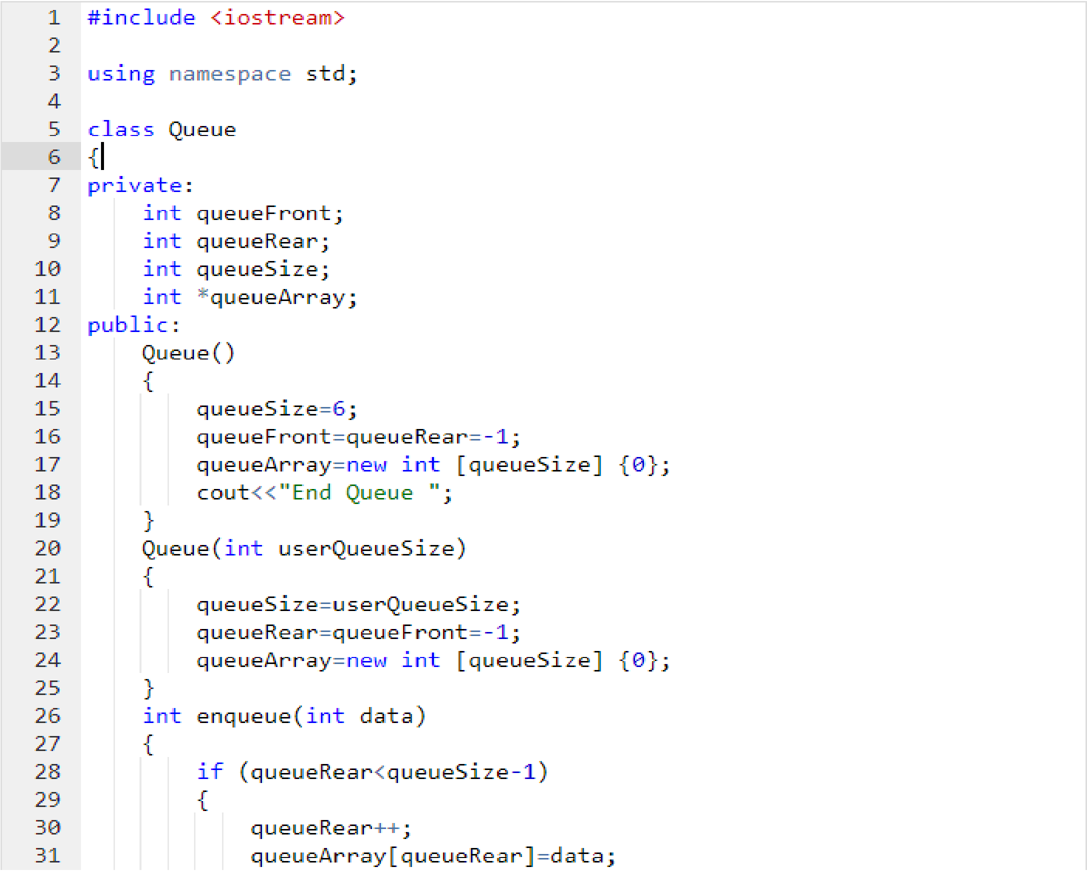


```cpp
}; //end stack class :)
int main()
{
    Stack stack1;
    Stack stack2;
    stack1.popFromStack();
    stack1.pushIntoStack(1);
    stack1.pushIntoStack(5);
    stack1.pushIntoStack(6);
    //stack1.printFromStack();
    stack1.popFromStack();
    stack1.pushIntoStack(7);
    stack1.popFromStack();
    stack1.popFromStack();
    stack1.popFromStack();
    stack1.popFromStack();
    stack1.pushIntoStack(7);
    stack1.popFromStack();

    //stack1.pushIntoStack(20);
   // stack2.popFromStack();
    return 0;
}
//non empty
```

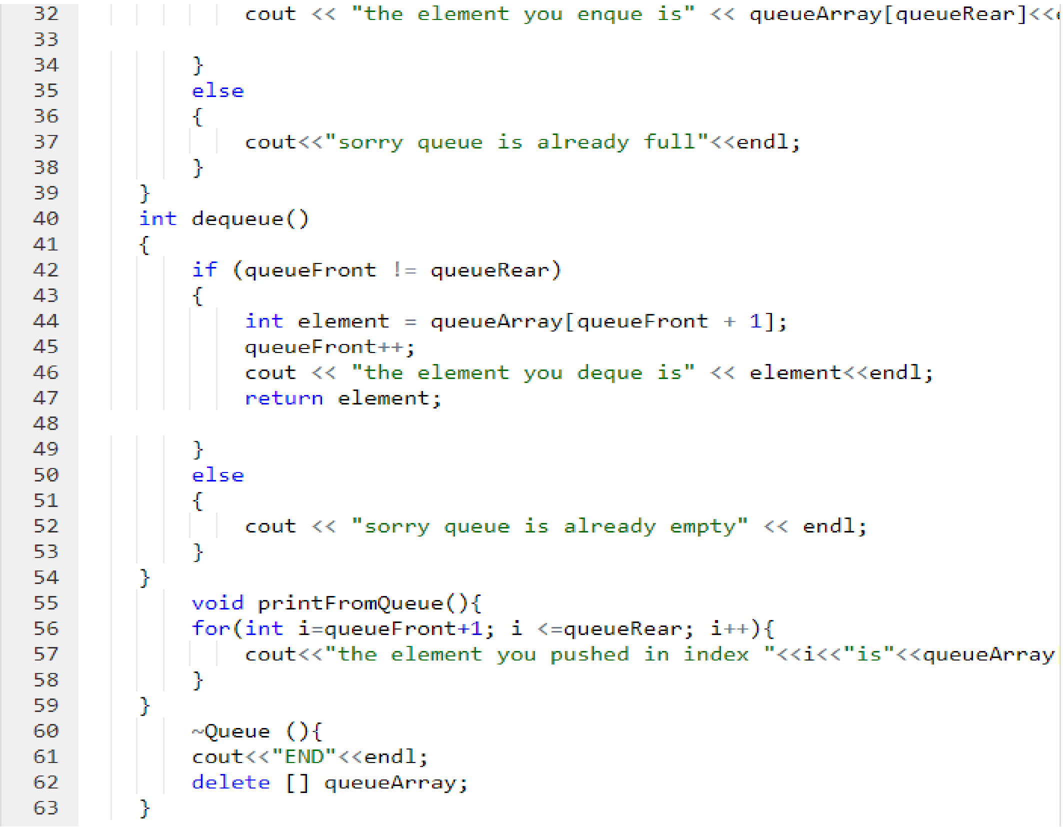


Print output (drag lower right corner to resize)

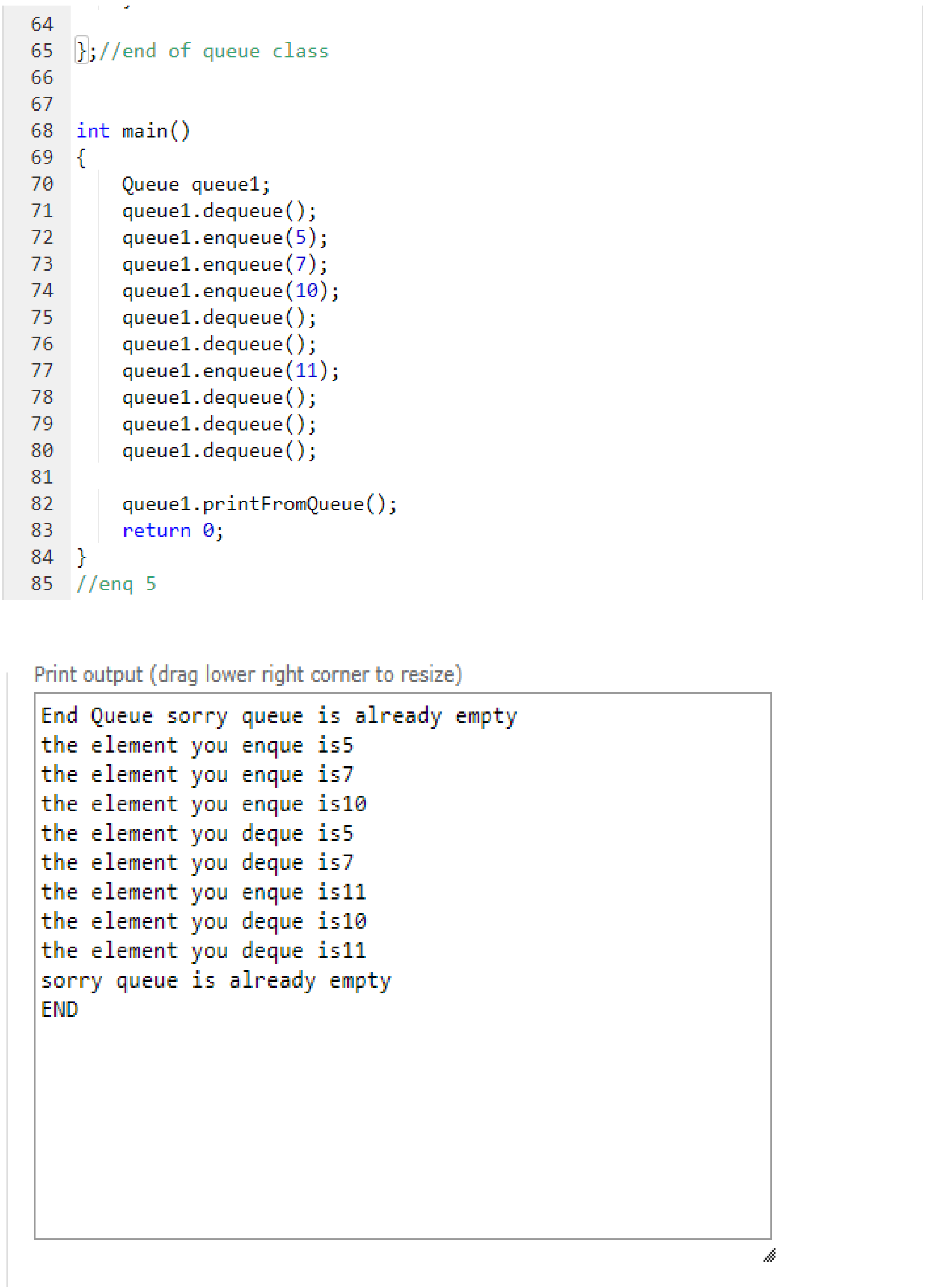
```
Sorry..the stack is already empty you can't pop from if
the element you pushed is 1
the element you pushed is 5
the element you pushed is 6
the element you poped is 6
the element you pushed is 7
the element you poped is 7
the element you poped is 5
the element you poped is 1
Sorry..the stack is already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



```cpp
#include <iostream>

using namespace std;

class Queue
{
private:
    int queueFront;
    int queueRear;
    int queueSize;
    int *queueArray;
public:
    Queue()
    {
        queueSize=6;
        queueFront=queueRear=-1;
        queueArray=new int [queueSize] {0};
        cout<<"End Queue ";
    }
    Queue(int userQueueSize)
    {
        queueSize=userQueueSize;
        queueRear=queueFront=-1;
        queueArray=new int [queueSize] {0};
    }
    int enqueue(int data)
    {
        if (queueRear<queueSize-1)
        {
            queueRear++;
            queueArray[queueRear]=data;
```



```cpp
            cout << "the element you enque is" << queueArray[queueRear]<<

        }
        else
        {
            cout<<"sorry queue is already full"<<endl;
        }
    }
    int dequeue()
    {
        if (queueFront != queueRear)
        {
            int element = queueArray[queueFront + 1];
            queueFront++;
            cout << "the element you deque is" << element<<endl;
            return element;

        }
        else
        {
            cout << "sorry queue is already empty" << endl;
        }
    }
        void printFromQueue(){
        for(int i=queueFront+1; i <=queueRear; i++){
            cout<<"the element you pushed in index "<<i<<"is"<<queueArray
        }
    }
        ~Queue (){
        cout<<"END"<<endl;
        delete [] queueArray;
    }
```



```cpp

};//end of queue class


int main()
{
    Queue queue1;
    queue1.dequeue();
    queue1.enqueue(5);
    queue1.enqueue(7);
    queue1.enqueue(10);
    queue1.dequeue();
    queue1.dequeue();
    queue1.enqueue(11);
    queue1.dequeue();
    queue1.dequeue();
    queue1.dequeue();

    queue1.printFromQueue();
    return 0;
}
//enq 5
```

```
End Queue sorry queue is already empty
the element you enque is5
the element you enque is7
the element you enque is10
the element you deque is5
the element you deque is7
the element you enque is11
the element you deque is10
the element you deque is11
sorry queue is already empty
END
```