

```

1  #include <iostream>
2
3  using namespace std;
4  class Stack
5  {
6  private :
7      int stackSize ;
8      int stackTop;
9      int *stackArray; |
10
11 public :
12     Stack()
13     {
14         stackSize=6;
15         stackTop=-1;
16         stackArray=new int [stackSize] {0};
17     }
18     Stack(int UserStackSize)
19     {
20         stackSize=UserStackSize;
21         stackTop=-1;
22         stackArray=new int [stackSize] {0};
23     }
24     int getTop(){
25         return stackTop;
26     }
27     int pushIntoStack (int data ){
28         if (stackTop<(stackSize-1)){
29             stackTop++;
30             stackArray[stackTop]=data;
31

```

```

32         cout<<"the element you pushed is "<<stackArray[stackTop]<<endl;
33     }
34     }
35     else {
36         cout<<"Sorry..the stack is already full you can't push into it";
37     }
38 }
39 int popFromStack(){
40     if (stackTop!=-1){
41         int element=stackArray[stackTop];
42         stackTop--;
43         cout<<"the element you popped is "<<element<<endl;
44         return element;
45     }
46     }
47     else {
48         cout<<"Sorry..the stack is already empty you can't pop from it";
49     }
50 }
51 void printFromStack(){
52     for(int i=0; i <=stackTop; i++){
53         cout<<"the element you pushed in index "<<i<<"is"<<stackArray[i]<<endl;
54     }
55 }
56 ~Stack (){
57     cout<<"END"<<endl;
58     delete [] stackArray;
59 }
60
61 }; //end stack class :)

```

```

61 }; //end stack class :)
62 int main()
63 {
64     Stack stack1;
65     Stack stack2;
66     stack1.popFromStack();
67     stack1.pushIntoStack(1);
68     stack1.pushIntoStack(5);
69     stack1.pushIntoStack(6);
70     //stack1.printFromStack();
71     stack1.popFromStack();
72     stack1.pushIntoStack(7);
73     stack1.popFromStack();
74     stack1.popFromStack();
75     stack1.popFromStack();
76     stack1.popFromStack();
77     stack1.pushIntoStack(7);
78     stack1.popFromStack();
79
80     //stack1.pushIntoStack(20);
81     // stack2.popFromStack();
82     return 0;
83 }
84 //end main

```

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

```

```

1  #include <iostream>
2
3  using namespace std;
4
5  class Queue
6  {
7  private:
8      int queueFront;
9      int queueRear;
10     int queueSize;
11     int *queueArray;
12 public:
13     Queue()
14     {
15         queueSize=6;
16         queueFront=queueRear=-1;
17         queueArray=new int [queueSize] {0};
18         cout<<"End Queue ";
19     }
20     Queue(int userQueueSize)
21     {
22         queueSize=userQueueSize;
23         queueRear=queueFront=-1;
24         queueArray=new int [queueSize] {0};
25     }
26     int enqueue(int data)
27     {
28         if (queueRear<queueSize-1)
29         {
30             queueRear++;
31             queueArray[queueRear]=data;
32
33             cout << "the element you enqueue is" << queueArray[queueRear]<<endl;
34         }
35         else
36         {
37             cout<<"sorry queue is already full"<<endl;
38         }
39     }
40     int dequeue()
41     {
42         if (queueFront != queueRear)
43         {
44             int element = queueArray[queueFront + 1];
45             queueFront++;
46             cout << "the element you deque is" << element<<endl;
47             return element;
48         }
49         else
50         {
51             cout << "sorry queue is already empty" << endl;
52         }
53     }
54     void printFromQueue(){
55         for(int i=queueFront+1; i <=queueRear; i++){
56             cout<<"the element you pushed in index "<<i<<"is"<<queueArray[i]<<endl;
57         }
58     }
59     ~Queue (){
60         cout<<"END"<<endl;
61         delete [] queueArray;
62     }
63 }

```

```

64
65 };//end of queue class
66
67
68 int main()
69 {
70     Queue queue1;
71     queue1.dequeue();
72     queue1.enqueue(5);
73     queue1.enqueue(7);
74     queue1.enqueue(10);
75     queue1.dequeue();
76     queue1.dequeue();
77     queue1.enqueue(11);
78     queue1.dequeue();
79     queue1.dequeue();
80     queue1.dequeue();
81
82     queue1.printFromQueue();
83     return 0;
84 }
85 //enq 5

```

Print output (drag lower right corner to resize)

```

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

```