```
1  public class Queue implements Interface
2  {
3      Object [] waitlist;
4      int front, back, count;
5
6      public Queue(int maxsize)
7      {
8          waitlist= new Object[maxsize];
9          front=0;
10         back=maxsize-1;
11         count=0;
12     }
13
14     public Queue()
15     {
16         this(50);
17     }
18
19     public void enqueue(Object x)
20     {
21         if(isFull()) return;
22         back++;
23         if(back>=waitlist.length) back=0;
24         waitlist[back]=x;
25         count++;
26     }
27
28     public Object dequeue()
29     {
30         if(isEmpty()) return null;
31         int oldfront=front;
32         front++;
33         if(front>=waitlist.length) front=0;
34         count=count-1;
35         return waitlist[oldfront];
36     }
37
38     public Object info()
39     {
40         if(isEmpty()) return null;
41         return waitlist[back];
42     }
43
44     public Object getFront()
45     {
46         return isEmpty()?null:waitlist[front];
47     }
48     public int size() {return count;}
49     public boolean isEmpty() {return count<=0;}
50     public boolean isFull() {return count>= waitlist.length;}
51     public void makeEmpty()
52     {
53         front=0;
54         back=waitlist.length-1;
55         count=0;
56     }
57
58 }
```