

```
1 import java.lang.*;
2
3 import javax.swing.plaf.basic.BasicTreeUI.TreeSelectionHandler;
4 public class Scoreboard{
5     public class GameEntry<Integer>{
6         private class Node<Integer>{
7             private int element;
8             private Node<Integer> prev;
9             private Node<Integer> next;
10            public Node(Integer e,Node<Integer> p,Node<Integer> n){
11                element = e;
12                prev = p;
13                next = n;
14            }
15            public Integer getElement(){return element;}
16            public Node<Integer> getPrev(){return prev;}
17            public Node<Integer> getNext(){return next;}
18            public void setPrev(Node<Integer> p){prev=p;}
19            public void setNext(Node<Integer> n){next=n;}
20        }
21        private Node<Integer> header;
22        private Node<Integer> trailer;
23        private int size=0;
24        public GameEntry(){
25            header = new Node<>(0,null,null);
26            trailer = new Node<>(0,header,null);
27            header.setNext(trailer);
28        }
29        private void addBetween(Integer e,Node<Integer> predecessor,Node<Integer> successor){
30            Node<Integer> newest = new Node<>(e,predecessor,successor);
31            predecessor.setNext(newest);
32            successor.setPrev(newest);
33            size++;
34        }
35        public void addLast(Integer e){
36            addBetween(e,trailer.getPrev(),trailer);
37        }
38        private Integer remove(Node<Integer> node){
39            Node<Integer> predecessor = node.getPrev();
40            Node<Integer> successor = node.getNext();
41            predecessor.setNext(successor);
42            successor.setPrev(predecessor);
43            size--;
44        }
45        public void adANDset(int k){
46            Node<Integer> p = trailer;
47            Node<Integer> q = p.getPrev();
48            if (size<10){
49                while(Integer.parseInt(p.getElement())<k){
50                    p=q;
51                    q=q.getPrev();
52                }
53                Integer.addBetween(k,p,p.getPrev());
54                size++;
55            }
56            else{
57                while(Integer.parseInt(p.getElement())<k){
58                    p=q;
59                    q=q.getPrev();
60                }
61                Integer.addBetween(k,p,p.getPrev());
62                Integer.remove(trailer);
63            }
64        }
65    }
66    Run | Debug
67    public static void main(String[] args){
68        GameEntry dlist=new GameEntry();
69        for (k=0;k<10;k++){
70            dlist.adANDset(k);
71        }
72        dlist.adANDset(k: 15);
73        dlist.adANDset(k: 3);
74        dlist.adANDset(k: 100);
75        dlist.adANDset(k: 4);
76        System.out.println(dlist);
77    }
78 }
```