



搜狐员工小王最近利用假期在外地旅游，在某个小镇碰到一个马戏团表演，精彩的表演结束后发现团长正和大伙在帐篷前激烈讨论，小王打听了下了解到，马戏团正打算出一个新节目“最高罗汉塔”，即马戏团员叠罗汉表演。考虑到安全因素，要求叠罗汉过程中，站在某个人肩膀上的人应该既比自己矮又比自己瘦，或相等。团长想要本次节目中的罗汉塔叠的最高，由于人数众多，正在头疼如何安排人员的问题。小王觉得这个问题很简单，于是统计了参与最高罗汉塔表演的所有团员的身高体重，并且很快找到叠最高罗汉塔的人员序列。现在你手上也拿到了这样一份身高体重表，请找出可以叠出的最高罗汉塔的高度，这份表中马戏团员依次编号为 1 到 N。

```
1  import java.util.Scanner;
2
3  public class Main {
4      public static class Dis{
5          int Num;           //马戏团成员的编号
6          int high;          //身高
7          int weight;        //体重
8          int max_high;      //记录这个马戏团成员为最下面的一个人，最多可以叠
9  多少层罗汉
10     }
11
12     public static void main(String args[]){
13         Scanner cin = new
14 Scanner(System.in);
15         while(cin.hasNext()){
16             int n = cin.nextInt();
17             Dis map[] = new Dis[n];
18             for(int i = 0;i < n;i++){
19                 map[i] = new Dis();           //每次进入的元素插入
20 队尾
21                 map[i].Num = cin.nextInt();
22                 map[i].weight = cin.nextInt();
23                 map[i].high = cin.nextInt();
24                 for(int j = i;j > 0;j--){      //使用冒泡排
25 序，对新插入的元素插入队列，按照体重从小到大的顺序排序
26                     if(map[j].weight < map[j-1].weight){
27                         int Num = map[j].Num;
28                         int high = map[j].high;
29                         int weight = map[j].weight;
30                         map[j].Num = map[j-1].Num;
31                         map[j].high = map[j-1].high;
32                         map[j].weight =
33  map[j-1].weight;
34                         map[j-1].Num = Num;
35                         map[j-1].high = high;
```



```
36         map[j-1].weight = weight;
37     }else if(map[j].weight ==
38     map[j-1].weight &&map[j].high > map[j-1].high){ //如果体重相同，身高矮的在后面
39         int Num = map[j].Num;
40         int high = map[j].high;
41         int weight = map[j].weight;
42         map[j].Num = map[j-1].Num;
43         map[j].high = map[j-1].high;
44         map[j].weight =
45     map[j-1].weight;
46         map[j-1].Num = Num;
47         map[j-1].high = high;
48         map[j-1].weight = weight;
49     }else
50         break; //队列已经有序了，跳
51     出循环
52     }
53 }
54     int max_high = getMaxHigh(map,n);
55     System.out.println(max_high);
56 }
57 }
58
59     private static int getMaxHigh(Dis[] map, int n) {
60         // TODO Auto-generated method stub
61         int max_high = 0;
62         for(int i = 0;i < n;i++){
63             map[i].max_high = 1;
64             for(int j = 0; j < i;j++){
65                 if(map[i].high >= map[j].high && map[i].max_high <
map[j].max_high+1){
66                     map[i].max_high = map[j].max_high + 1;
67                 }
68             }
69             max_high = Math.max(max_high, map[i].max_high);
70         }
71         return max_high;
72     }
73 }
```

两个搜狐的程序员加了一个月班，终于放假了，于是他们决定扎金花渡过愉快的假期。



游戏规则：

共 52 张普通牌，牌面为 2,3,4,5,6,7,8,9,10,J,Q,K,A 之一，大小递增，各四张； 每人抓三张牌。两人比较手中三张牌大小，大的人获胜。

对于牌型的规则如下：

1. 三张牌一样即为豹子
2. 三张牌相连为顺子（A23 不算顺子）
3. 有且仅有两张牌一样为对子 豹子>顺子>对子>普通牌型 在牌型一样时，比较牌型数值大小（如 AAA>KKK, QAK>534, QQ2>10104） 在二人均无特殊牌型时，依次比较三张牌中最大的。大的人获胜，如果最大的牌一样，则比较第二大，以此类推（如 37K>89Q） 如二人牌面相同，则为平局。

```
1      /*
2
3      * 对于扑克牌,将不在附近的 JQKA 和 10, 交换成 IJKLM<=>10, J, Q, K, A, 则替换后非常
4      好处理.
5
6      */
7
8      #include <iostream>
9
10     #include <string>
11
12     #include <algorithm>
13
14     using namespace std;
15
16
17
18     /* return a <type, first_max_element> */
19
20     pair<int, int> judgeType(string& s)
21
22     {
23
24         int len = s.size();
```





```
25
26         sort(s.begin(), s.end());
27
28         if(s[0] == s[1] && s[1] == s[2])
29
30             returnmake_pair(6, s[0]);    // KKK
31
32         elseif(s[1]-s[0] == 1&& s[2]-s[1] == 1)
33
34             returnmake_pair(5, s[0]);    // JQK
35
36         if(s[0] == s[1])
37
38             returnmake_pair(4, s[0]);    // JJA
39
40         elseif(s[0] == s[2])
41
42             returnmake_pair(4, s[0]);
43
44         elseif(s[1] == s[2])
45
46             returnmake_pair(4, s[1]);
47
48         returnmake_pair(3, *max_element(s.begin(), s.end()));
49     }
50
51
52
53
54     string& exchange(string& raw, string ns, string ne)
55
56     {
57
58         intlen = raw.size();
59
60         intp1 = raw.find(ns);
61
62         while(p1 < len && p1 >= 0)
63
64             {
65
66                 raw.replace(raw.begin()+p1, raw.begin()+p1+ns.size(), ne);
67
68                 p1 = raw.find(ns, p1);
```





```
69
70     }
71
72     return raw;
73
74 }
75
76
77
78
79
80 int main()
81 {
82
83     string s1;
84
85     string s2;
86
87     while (cin >> s1 >> s2)
88     {
89
90         // I-J-K-L-M-N <=> 10, J, Q, K, A
91
92         s1 = exchange(s1, "10", "I");
93
94         s1 = exchange(s1, "K", "L");
95
96         s1 = exchange(s1, "Q", "K");
97
98         s1 = exchange(s1, "A", "M");
99
100        s2 = exchange(s2, "10", "I");
101
102        s2 = exchange(s2, "K", "L");
103
104        s2 = exchange(s2, "Q", "K");
105
106        s2 = exchange(s2, "A", "M");
107
108        pair<int, int> t1 = judgeType(s1);
109
110        pair<int, int> t2 = judgeType(s2);
111
112    }
```

获取更多资料礼包!

微信关注: 白熊事务所



```
113
114         if(s1 == s2)
115
116             cout << "0"<< endl;
117
118         else
119
120         {
121
122             if(t1.first > t2.first)
123
124                 cout << "1"<< endl;
125
126             elseif(t1.first < t2.first)
127
128                 cout << "-1"<< endl;
129
130             else
131
132             {
133
134                 if(t2.second != t1.second)
135
136                     cout << (t1.second >
137 t2.second ? "1": "-1") << endl;
138
139                     elseif(s1[1] != s2[1])
140
141                         cout << (s1[1] > s2[1] ? "1": "-1")
142 << endl;
143
144                     elseif(s1[0] != s2[0])
145
146                         cout << (s1[0] > s2[0] ? "1": "-1")
147 << endl;
148
149                     else
150
151                         cout << "0"<< endl;
152
153             }
154
155         }
156
```



157

}

return 0;

}

狐进行了一次黑客马拉松大赛，全公司一共分为了 N 个组，每组一个房间排成一排开始比赛，比赛结束后没有公布成绩，但是每个组能够看到自己相邻的两个组里比自己成绩低的组的成绩，比赛结束之后要发奖金，以 $1w$ 为单位，每个组都至少会发 $1w$ 的奖金，另外，如果一个组发现自己的奖金没有高于比自己成绩低的组发的奖金，就会不满意，作为比赛的组织方，根据成绩计算出至少需要发多少奖金才能让所有的组满意。

```
1  import java.util.Scanner;
2  public class Main {
3      public static void main(String[] args) {
4          Scanner scanner = new Scanner(System.in);
5          while (scanner.hasNext()) {
6              int N = scanner.nextInt();
7              int[] grades = new int[N];
8              for (int i = 0; i < N; i++) {
9                  grades[i] = scanner.nextInt();
10             }
11
12             int[] bonus = new int[N];
13             int[] cobonus = new int[N];
14             bonus[0] = 1;
15             cobonus[N-1] = 1;
16             for (int i = 1; i < grades.length; i++) {
17                 if (grades[i] > grades[i-1])
18                     bonus[i] = bonus[i-1] + 1;
19                 else
20                     bonus[i] = 1;
21             }
22             for (int i = N-1; i > 0; i--) {
23                 if (grades[i-1] > grades[i])
24                     cobonus[i-1] = cobonus[i] + 1;
25                 else
26                     cobonus[i-1] = 1;
27             }
28
29             int sum = 0;
```



```
30             for (int i = 0; i < N; i++) {
31                 int temp =
32     bonus[i]>cobonus[i]?bonus[i]:cobonus[i];
33                 sum += temp;
34             }
35             System.out.println(sum);
36         }
37     }
```

从头到尾，从尾到头都来一遍，结果分别存在两个数组里，最后取两个数组中米一个元素较大的一个，相加得到最后结果。

