



挑选特种兵

【问题描述】

近期有伙极端分子占据了一个小岛，为了打击这伙极端势力，指挥部要派一名特种兵前去侦察，为此，进行了一场多项军事科目的比赛。

指挥部要对参赛选手先后参加的泅渡、野外生存、射击等成绩进行评分，选出各科目的冠军。然后从冠军中间选出三科目最差成绩最高的选手，安排他前去执行任务。

如果有多个选手成绩相同，则选序号最小的选手。。

【输入格式】

第一行是一个正整数 n ，表示选手人数 ($n \leq 200$)。

下面 n 行，第 i 行是序号为 $i-1$ 的选手的泅渡、野外生存、射击的科目成绩（整数， ≤ 100 分）。

【输出格式】

有 1 行，输出执行任务的特种兵序号。

【输入样例】

```
5
89 98 99
98 98 98
100 89 95
95 95 93
97 91 100
```

【输出样例】

```
2
```

【分析】

根据比赛规则，先把各科的冠军选出来，加入到种子选手队伍中

然后根据最差成绩最高的规则选。

如果最差成绩相同，则选择序号小的选手。

**【参考程序】**

```
#include<iostream>
#include<algorithm>
using namespace std;
const int N = 200;
struct SF
{
    int id; //序号
    int score1; //泗渡
    int score2; //野外生存
    int score3; //射击
    int min_score; //最低成绩
};
SF sf[N + 1]; //特种兵
SF sf_best[3 * N + 1]; //各科冠军
bool cmp1(SF a, SF b);
bool cmp2(SF a, SF b);
bool cmp3(SF a, SF b);
bool cmp4(SF a, SF b);
int main()
{
    int n;
    cin >> n;
    //读取选手成绩
    for(int i = 1; i <= n; i++)
    {
        sf[i].id = i;
        cin >> sf[i].score1;
        cin >> sf[i].score2;
        cin >> sf[i].score3;
        sf[i].min_score = min(sf[i].score1, min(sf[i].score2, sf[i].score3));
    }
    //选出泗渡冠军
    sort(sf + 1, sf + n + 1, cmp1);
    int k = 1; //种子选手计数
    //把泗渡冠军加入到种子选手中
    for(int i = 1; i <= n; i++)
    {
        if(sf[i].score1 != sf[1].score1)
        {
            break;
        }
    }
```



```
        sf_best[k] = sf[i];
        k++;
    }
    //选出野外生存冠军
    sort(sf + 1, sf + n + 1, cmp2);
    //把野外生存冠军加入到种子选手中
    for(int i = 1; i <= n; i++)
    {
        if(sf[i].score2 != sf[1].score2)
        {
            break;
        }
        sf_best[k] = sf[i];
        k++;
    }
    //选出射击冠军
    sort(sf + 1, sf + n + 1, cmp3);
    //把射击冠军加入到种子选手中
    for(int i = 1; i <= n; i++)
    {
        if(sf[i].score3 != sf[1].score3)
        {
            break;
        }
        sf_best[k] = sf[i];
        k++;
    }
    //根据最差成绩排序，选出没有弱项的选手
    sort(sf_best + 1, sf_best + k, cmp4);
    cout << sf_best[1].id << endl;
    return 0;
}
//选出泅渡冠军
bool cmp1(SF a, SF b)
{
    return a.score1 > b.score1;
}
//选出野外生存冠军
bool cmp2(SF a, SF b)
{
    return a.score2 > b.score2;
}
//选出射击冠军
bool cmp3(SF a, SF b)
```



```
{  
    return a.score3 > b.score3;  
}  
//选出最差成绩最高的选手  
bool cmp4(SF a, SF b)  
{  
    //先选择最差成绩高的  
    if(a.min_score != b.min_score)  
    {  
        return a.min_score > b.min_score;  
    }  
    //最差成绩相同，根据序号从小到大排序  
    return a.id < b.id;  
}
```