

☑ PAT 乙级题目讲解：1015 《德才论》

🔗 题目简介

本题考查结构体排序与多条件比较，是一道典型的“模拟+自定义排序”问题。题目源自宋代司马光提出的“德才论”，要求我们依据德分和才分将考生分类排序。

考生被分为以下四类：

1. **才德全尽（圣人）**：德分和才分均不小于优先线 H ；
2. **德胜才（君子）**：德分 $\geq H$ ，才分 $< H$ ；
3. **才德兼亡但德胜才（小人）**：德分 $< H$ 且才分 $< H$ ，但德分 \geq 才分；
4. **其余及格者（愚人）**：其余德分 $\geq L$ ，才分 $\geq L$

排序优先级：

总分从高到低 \rightarrow 德分从高到低 \rightarrow 准考证号从小到大。（三级嵌套）

🔗 样例分析

样例输入：

```
14 60 80
10000001 64 90
10000002 90 60
...
```

总共 14 个学生，最低线 L 为 60，优先线 H 为 80。遍历每个考生后：

- 满足德才 ≥ 60 的为合格考生，计数人数 m ；
- 再按分数划分四类，分别加入对应数组；
- 每类内部排序后依次输出所有人。

输出：

```
12
10000013 90 99
10000012 80 100
...
```

🔍 解题思路

本题关键是：**考生分类 + 多维排序**，理解四类考生分类标准如下。

类别	条件	优先级
第 1 类（才德全尽）	德分 $\geq H$ 且 才分 $\geq H$	最高
第 2 类（德胜才）	德分 $\geq H$ 且 才分 $< H$	第二
第 3 类（德才兼亡但德胜才）	德分 $< H$ 且 才分 $< H$ 且 德分 \geq 才分	第三
第 4 类（普通达线）	德才 $\geq L$ 但不满足前几类	最低



变量说明

变量名	含义
<code>n</code>	考生总数
<code>l</code>	录取最低分数线
<code>h</code>	优先录取分数线
<code>m</code>	合格考生总数
<code>id</code>	准考证号
<code>d</code>	德分
<code>c</code>	才分
<code>s</code>	总分
<code>a1 ~ a4</code>	四类考生数组
<code>c1 ~ c4</code>	每类考生人数计数器
<code>t</code>	临时学生对象



Step 1: 定义结构体并读入数据

```
struct Stu {
    int id, d, c, s;
} a1[maxn], a2[maxn], a3[maxn], a4[maxn], t;
```

读取 `n` 个考生数据，并判断是否达到最低线：

```
if(t.d >= l && t.c >= l){
    m++;
    // 分类进入 a1~a4
}
```



Step 2: 对每位考生判断分类，分组保存

我们需将所有达线（德分和才分均 $\geq L$ ）的考生，按照上述条件放入不同数组 `a1 ~ a4`。

```
if(t.d >= h && t.c >= h)    → 存入 a1, c1 记录人数
else if(t.d >= h && t.c < h) → 存入 a2, c2 记录人数
else if(t.d >= t.c)         → 存入 a3, c3 记录人数
else                       → 存入 a4, c4 记录人数
```

☑ Step 3: 排序规则设计

每一类内部的排序规则相同，按以下顺序：

1. 德才总分降序；
2. 若总分相同，德分高者优先；
3. 若德分也相同，准考证号小者优先。

可以使用自定义排序函数处理，比较规则函数 `cmp` 定义如下。

```
bool cmp(Stu x, Stu y){
    if(x.s == y.s){ // 若总分相同
        if(x.d == y.d) // 若德分也相同
            return x.id < y.id; // 准考证号小者优先
        return x.d > y.d; // 德分高者优先
    }
    return x.s > y.s; // 德才总分降序
}
```

☑ Step 4: 合并所有类别结果并输出

四类考生分别排序后依次合并，输出最终排序结果。

```
sort(a1 + 1, a1 + c1 + 1, cmp);
sort(a2 + 1, a2 + c2 + 1, cmp);
sort(a3 + 1, a3 + c3 + 1, cmp);
sort(a4 + 1, a4 + c4 + 1, cmp);

printf("%d\n", m);
...
```

☑ 完整代码

```
#include<bits/stdc++.h>
using namespace std;

const int maxn = 1e5 + 5;
int n, l, h, m, c1, c2, c3, c4;

struct Stu{
    int id, d, c, s;
}a1[maxn], a2[maxn], a3[maxn], a4[maxn], t;

bool cmp(Stu x, Stu y){
    if(x.s == y.s){ // 若总分相同
        if(x.d == y.d) // 若德分也相同
```

```

        return x.id < y.id; // 准考证号小者优先
        return x.d > y.d; // 德分高者优先
    }
    return x.s > y.s; // 德才总分降序
}

int main(){
    scanf("%d %d %d", &n, &l, &h);
    for(int i = 1; i <= n; i++){
        scanf("%d %d %d", &t.id, &t.d, &t.c);
        t.s = t.d + t.c;
        if(t.d >= 1 && t.c >= 1){
            m++;
            if(t.d >= h && t.c >= h){
                a1[++c1] = t;
            }
            else if(t.c < h && t.d >= h){
                a2[++c2] = t;
            }
            else if(t.d >= t.c){
                a3[++c3] = t;
            }
            else{
                a4[++c4] = t;
            }
        }
    }
    sort(a1 + 1, a1 + c1 + 1, cmp);
    sort(a2 + 1, a2 + c2 + 1, cmp);
    sort(a3 + 1, a3 + c3 + 1, cmp);
    sort(a4 + 1, a4 + c4 + 1, cmp);
    printf("%d\n", m);
    for(int i = 1; i <= c1; i++){
        printf("%d %d %d\n", a1[i].id, a1[i].d, a1[i].c);
    }
    for(int i = 1; i <= c2; i++){
        printf("%d %d %d\n", a2[i].id, a2[i].d, a2[i].c);
    }
    for(int i = 1; i <= c3; i++){
        printf("%d %d %d\n", a3[i].id, a3[i].d, a3[i].c);
    }
    for(int i = 1; i <= c4; i++){
        printf("%d %d %d", a4[i].id, a4[i].d, a4[i].c);
        if(i < c4) printf("\n");
    }
    return 0;
}

```

🚫 常见错误提醒

错误点	原因说明
分类判断顺序写错	顺序错误会导致学生进错组
sort 比较函数条件遗漏	总分、德分、ID 三重判断缺一不可
最后一类输出多余换行	特判最后一位避免尾部多输出

☑ 总结归纳

- 本题是**多维排序 + 多组存储**的经典模拟题；
- 分类优先级需与题意严格一致；
- 排序需覆盖三重维度，总分 > 德分 > 准考证号；
- 四类考生独立排序后统一输出；

时间复杂度：

- 读入与分类： $O(n)$ ；
- 排序： $O(n \log n)$ （最多四次排序，sort 内部实现是快排）；
- 输出： $O(n)$ ；
- 总体： $O(n \log n)$

空间复杂度：

- $O(n)$ ：数组存储所有合格考生数据；

🧠 思维拓展

- 本题分类排序思想可迁移至：成绩评比、赛事排名、资源调度等问题；
- 若数据量更大，需考虑使用堆或归并优化排序部分；
- 若考生分类进一步增加，可考虑用 vector 数组结构替代四个静态数组。