

# 成绩排序

C 语言课程的上机考试有  $M$  道题目，从第 1 题至第  $M$  题每道题的分值均为大于 0 的正整数  $a_1, a_2, \dots, a_m$  ( $a_1, a_2, \dots, a_m$  均为大于 0 的正整数)。

现给定分数线，请编写程序输出不低于该分数线的学生人数，并将他(她)们的成绩按**降序**输出，若有多名学生考试成绩相同，则按他(她)们学号的**升序**输出。

## 输入

第 1 行是 3 个正整数：学生人数  $N$  ( $0 < N < 500$ )、考试题目数  $M$  ( $0 < M \leq 10$ )、分数线  $G$  ( $G > 0$ )；

第 2 行有  $M$  个正整数：分别为第 1 题至第  $M$  题的分值  $a_1, a_2, \dots, a_m$  ( $0 < a_x \leq 10000000$ )；

接下来  $N$  行，每行给出一名学生的学号（长度为 11 的字符串，学号的第一个字符可能为 0）、该学生解出的题目总数  $S$  ( $S \geq 0$ )、以及这  $S$  道题的题号（题号由 1 到  $M$ ）。

## 输出

然后首先在一行中输出不低于分数线的学生人数。

接下来每行按考试成绩从高到低输出上分数线的学生的学号与成绩，其间用 1 个空格分隔，若有多名学生成绩相同，则按他(她)们学号的升序输出。

## 样例1

```
1 4 5 25
2 10 10 12 13 15
3 10091130015 3 5 1 3
4 10091130013 5 2 4 1 3 5
5 10091130012 2 1 2
6 10091130011 3 2 3 5
```

考试有 5 题，第 1 题 10 分，第 2 题 10 分，第 3 题 12 分，第 4 题 13 分，第 5 题 15 分。

考生 10091130015 做出 3 题，分别为 5、1、3，所以得分  $15+10+12=37$  分

考生 10091130013 做出 5 题，分别为 2、4、1、3、5，所以得分 60 分

同理可得其他考生分数

及格线 25 分，所以有 3 人及格，分别是 10091130013 (60 分)、10091130011 (37 分) 和 10091130015 (37 分)

首先按照分数高低排序，发现同分的有 2 人，则这两个人再按照学号排序，11 排在 15 前面  
故输出为：

```
1 10091130013 60
2 10091130011 37
3 10091130015 37
```

## 样例2

```
1 2 3 20
2 10 10 10
3 10101130012 0
4 10101130019 2 1 2
```

考试有 3 题，都是 10 分

10101130012 做出 0 题，是 0 分

10101130019 做出 2 题，分别是 1、2，所以得到 10+10=20 分

及格线是 20 分，所以及格的有 1 人，为 10101130019，是 20 分

```
1 1
2 10101130019 20
```

### 样例3

```
1 1 2 40
2 10 30
3 10101130018 1 2
```

没有人及格

```
1 0
```

## 可使用的模板

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <malloc.h>
4 #include <string.h>
5
6 struct student {
7     char name[100];
8     int score;
9 };
10
11 int cmp(const void*a, const void*b) {
12     struct student* pa = (struct student*)a;
13     struct student aa = *pa;
14
15     struct student* pb = (struct student*)b;
16
17     if (aa.score > pb->score) {
18         return -1;
19     } else if (aa.score == pb->score) {
20         return strcmp(aa.name, pb->name);
21     } else {
22         return 1;
23     }
24
25 }
26
27 int main()
28 {
```

```
29     struct student A[4];
30
31     int i;
32     for (i=0;i<4;i++) {
33         char y[100];
34         scanf("%s", y);
35         int x;
36         scanf("%d", &x);
37         A[i].score = x;
38         strcpy(A[i].name, y);
39     }
40
41     qsort(A, 4, sizeof(struct student), cmp);
42
43     for (i=0;i<4;i++) {
44         printf("student #%d:", i);
45         printf("name=%s,", A[i].name);
46         printf("score=%d\n", A[i].score);
47     }
48
49
50     return 0;
51
52 }
53
```