

1. 现有一份用户名单，你需要找出得分最高的用户。用户名单中含有以下信息：

- **name**：一个长度在 $[1,15]$ 范围内的字符串，字符串中只含有大小写字母、数字和下划线，当然不同用户的 **name** 可能重复。
- **id**：一个在 $[1,10^7]$ 范围内的整数，**id** 具有唯一性。
- **score**：一个在 $[-9999,9999]$ 范围内的整数，不同用户的 **score** 可能重复。

当有多个 **score** 最高的用户时，输出 **id** 最小的。

输入格式

第一行一个整数 n ($1 \leq n \leq 10^5$)，代表用户个数。

接下来的 n 行，每行一个字符串 **name** 和两个整数 **id**，**score**，以空格分隔。

输出格式

输出一行，一个字符串和两个整数，以空格分隔，表示 **score** 最高的用户的 **name**，**id** 和 **score**。

输入样例 1

```
6
SoaringDream 5455073 233
shorn1 873974 1500
Yukimai 2282197 2161
jqe 1 2161
Dev_XYS 2 1648
RA 3 -8000
```

输出样例

```
jqe 1 2161
```

```
1 #include<stdio.h>
2 #include<string.h>
3 struct Stu{
4     char name[10];
5     int id;
6     int score;
7 }stu[100000];
8
9 int main()
10 {
11     int N;
12     int i,max;
13     scanf("%d",&N);
14     for(i=0;i<N;i++){
15         scanf("%s%d%d",stu[i].name,&stu[i].id,&stu[i].score);
16         if(i==0)max=i;
17         else if(stu[i].score>stu[max].score)max=i;
18         else if(stu[i].score==stu[max].score&&stu[i].id<stu[max].id)max=i;
19     }
20     printf("%s %d %d",stu[max].name,stu[max].id,stu[max].score);
21     return 0;
22 }
```

6

SoaringDream 5455073 233

shorn1 873974 1500

Yukimai 2282197 2161

jqe 1 2161

Dev_XYS 2 1648

RA 3 -8000

jqe 1 2161

Run Succeeded Time 242 ms Peak Memory 10.2M

2. 请将第一题的输入从键盘改至文件读入， 输出数据保存至某一磁盘文件中。

```
1 #include<stdio.h>
2 #include<string.h>
3 #include<stdlib.h>
4 struct Stu
5 {
6     char name[10];
7     int id;
8     int score;
9 }stu[100000];
10 int main ()
11 {
12     FILE *fp;
13     int N;
14     int i,max;
15     if((fp=fopen("in.txt","r"))==NULL)exit(EXIT_FAILURE);
16     fscanf(fp,"%d",&N);
17     for(i=0;i<N;i++){
18         fscanf(fp,"%s%d%d",stu[i].name,&stu[i].id,&stu[i].score);
19         if(i==0)max=i;
20         else if(stu[i].score>stu[max].score)max=i;
21         else if(stu[i].score==stu[max].score&&stu[i].id<stu[max].id)max=i;
22     }
23     fclose(fp);
24     if((fp=fopen("out.txt","w"))==NULL)exit(EXIT_FAILURE);
25     fprintf(fp,"%s %d %d",stu[max].name,stu[max].id,stu[max].score);
26     fclose(fp);
27     return 0;
28 }
```

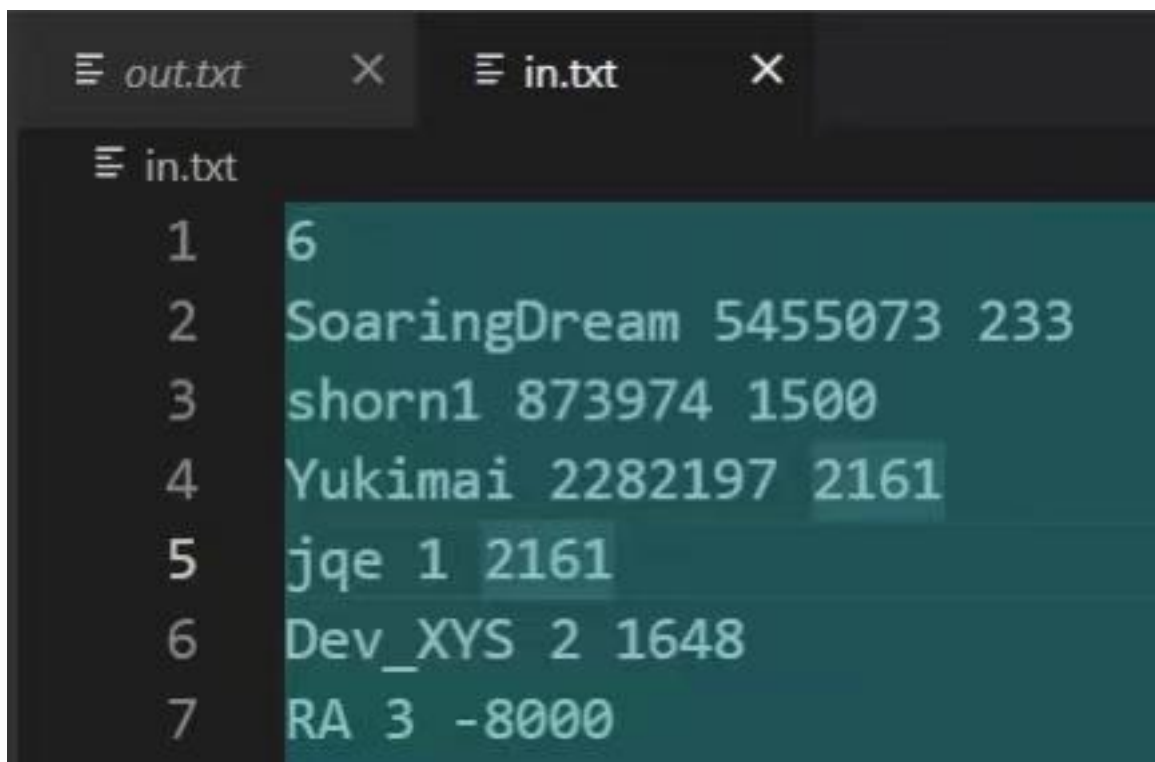
Symbol | Tabs: 4 | Line 1, Column 1

```
out.txt x in.txt x
in.txt
1 6
2 SoaringDream 5455073 233
3 shorn1 873974 1500
4 Yukimai 2282197 2161
5 jqe 1 2161
6 Dev_XYS 2 1648
7 RA 3 -8000
```

```
out.txt x in.txt
out.txt
1 jqe 1 2161
```

3. 请将第一题的用户名单按照score 降序排列输出某一磁盘文件中， 若 score 相同， id 号小的在前。

```
C
Language Run Stop Run Settings...
71066001-陈伟杰-上机7.3.c
1 #include<stdio.h>
2 #include<string.h>
3 #include<stdlib.h>
4 struct Stu
5 {
6     char name[16];
7     int id;
8     int score;
9 }stu[100000],t;
10 int main()
11 {
12     FILE *fp;
13     int N;
14     int i,j,flag;
15     if((fp=fopen("in.txt","r"))==NULL)exit(EXIT_FAILURE);
16     fscanf(fp,"%d",&N);
17     for(i=0;i<N;i++){
18         fscanf(fp,"%s%d%d",&stu[i].name,&stu[i].id,&stu[i].score);
19     }
20
21     fclose(fp);
22     for(i=0;i<N;i++){
23         flag=1;
24         for(j=i+1;j<N;j++){
25             if(stu[j-1].score<stu[j].score){
26                 flag=0;
27                 t=stu[j-1];
28                 stu[j-1]=stu[j];
29                 stu[j]=t;
30             }
31             else if(stu[j-1].score==stu[j].score&&stu[j-1].id>stu[j].id){flag=0;
32                 t=stu[j-1]; stu[j-1]=stu[j];
33                 stu[j]=t;
34             }
35         }
36         if(flag)break;
37     }
38     if((fp=fopen("out.txt","w"))==NULL)exit(EXIT_FAILURE);
39     for(i=0;i<N;i++){
40         fprintf(fp,"%s %d %d\n",stu[i].name,stu[i].id,stu[i].score);
41     }
42     fclose(fp);
43     return 0;
44 }
```



≡ in.txt

≡ out.txt

×

≡ out.txt

```
1 jqe 1 2161
2 Yukimai 2282197 2161
3 Dev_XYS 2 1648
4 shorn1 873974 1500
5 SoaringDream 5455073 233
6 RA 3 -8000
7
```