排序输出排名，同分同排名的做法

在pat的题库里，会有很多要分数相同时排名不变的要求。记录一个自己常用的函数

```
vector<stu> setRank(vector<stu> temp1,int room)
{
    int cnt1 = 0, cnt2 = 0;
    while (cnt1 != room)
    {
        if (temp1[cnt1].score == temp1[cnt2].score) temp1[cnt1++].rank = cnt2 +
1;
        else cnt2 = cnt1;
    }
    return temp1;
}
```

看起来还是很简洁的哈哈哈。能够按照要求，同分同排名。

留一道例题：

1025 PAT Ranking (25分)

Programming Ability Test (PAT) is organized by the College of Computer Science and Technology of Zhejiang University. Each test is supposed to run simultaneously in several places, and the ranklists will be merged immediately after the test. Now it is your job to write a program to correctly merge all the ranklists and generate the final rank.

Input Specification:

Each input file contains one test case. For each case, the first line contains a positive number *N* (≤100), the number of test locations. Then *N* ranklists follow, each starts with a line containing a positive integer *K* (≤300), the number of testees, and then *K* lines containing the registration number (a 13-digit number) and the total score of each testee. All the numbers in a line are separated by a space.

Output Specification:

For each test case, first print in one line the total number of testees. Then print the final ranklist in the following format:

```
registration_number final_rank location_number local_rank
```

The locations are numbered from 1 to *N*. The output must be sorted in nondecreasing order of the final ranks. The testees with the same score must have the same rank, and the output must be sorted in nondecreasing order of their registration numbers.

Sample Input:

```
2
5
1234567890001 95
1234567890005 100
1234567890003 95
1234567890002 77
1234567890004 85
4
1234567890013 65
1234567890011 25
1234567890014 100
1234567890012 85
```

Sample Output:

```
9
1234567890005 1 1 1
1234567890014 1 2 1
1234567890001 3 1 2
1234567890003 3 1 2
1234567890004 5 1 4
1234567890012 5 2 2
1234567890002 7 1 5
1234567890013 8 2 3
1234567890011 9 2 4
```

很简单的，用一个结构体存东西，用sort和上面的函数分级排序和总排序就搞定了

代码参考

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <string>
using namespace std;

struct stu
{
    string id;
    int loca, score, rank, rankG;
};
vector<stu> pat;
bool cmp(stu a,stu b)
{
    if (a.score == b.score)
        return a.id < b.id;
    return a.score > b.score;
}
vector<stu> setRank(vector<stu> temp1,int room)
{
    int cnt1 = 0, cnt2 = 0;
    while (cnt1 != room)
    {
```

```
        if (temp1[cnt1].score == temp1[cnt2].score) temp1[cnt1++].rank = cnt2 +
1;
        else cnt2 = cnt1;
    }
    return temp1;
}
vector<stu> setRankG(vector<stu> temp1, int room)
{
    int cnt1 = 0, cnt2 = 0;
    while (cnt1 != room)
    {
        if (temp1[cnt1].score == temp1[cnt2].score) temp1[cnt1++].rankG = cnt2 +
1;
        else cnt2 = cnt1;
    }
    return temp1;
}
int main()
{
    int location = 0, room = 0;
    cin >> location;
    for(int j=0;j<location;j++)
    {
        cin >> room;
        vector<stu> temp1;
        for (int i = 0; i < room; i++)
        {
            stu temp;
            cin >> temp.id >> temp.score;
            temp.loca = j + 1;
            temp1.push_back(temp);
        }
        sort(temp1.begin(), temp1.end(), cmp);
        temp1 = setRank(temp1, room);
        for (int i = 0; i < room; i++) pat.push_back(temp1[i]);
    }
    sort(pat.begin(), pat.end(), cmp);
    pat = setRankG(pat, pat.size());
    cout << pat.size() << endl;
    for (int i = 0; i < pat.size(); i++)
        cout << pat[i].id << " " << pat[i].rankG << " " << pat[i].loca << " " <<
pat[i].rank << endl;
    return 0;
}
```