

☑ PAT 乙级题目讲解：1004 《成绩排名》

🔗 题目简介

本题要求读入若干名学生的信息（姓名、学号、成绩），输出其中成绩最高和成绩最低的学生的姓名和学号。

核心任务是使用结构体封装每个学生的数据，并通过自定义排序函数来查找分数最高和最低的学生。

🔧 样例分析

输入：

```
3
Joe Math990112 89
Mike CS991301 100
Mary EE990830 95
```

分析过程：

- 三位学生成绩依次为 89、100、95
- 成绩最高的是 Mike，最低的是 Joe

因此输出为：

```
Mike CS991301
Joe Math990112
```

🔍 解题思路

👤 变量说明

变量名	含义
<code>n</code>	学生人数
<code>stu</code>	学生结构体数组
<code>nm</code>	学生姓名
<code>id</code>	学号
<code>s</code>	成绩
<code>cmp</code>	自定义排序函数，按成绩升序排序

本题的解决流程可以分为以下几个步骤：

☑ Step 1: 定义结构体

封装每个学生的信息，使用结构体 `Stu` 表示：

```
struct Stu{
    string nm, id;
    int s;
}stu[100005];
```

☑ Step 2: 定义自定义排序规则

按成绩从小到大排序（分数小的排前面）：

```
bool cmp(Stu x, Stu y){
    return x.s < y.s;
}
```

☑ Step 3: 输入数据

使用 `cin` 读入每个学生的姓名、学号和成绩：

```
for(int i = 1; i <= n; i++){
    cin >> stu[i].nm >> stu[i].id >> stu[i].s;
}
```

☑ Step 4: 排序并输出结果

使用 `sort` 对学生数组按成绩升序排序，然后输出最高（最后一位）和最低（第一位）：

```
sort(stu + 1, stu + 1 + n, cmp);

cout << stu[n].nm << " " << stu[n].id << "
";
cout << stu[1].nm << " " << stu[1].id;
```

完整代码

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

int n;
struct Stu{
    string nm, id;
    int s;
}stu[100005];

bool cmp(Stu x, Stu y){
    return x.s < y.s;
}
```

```
int main(){
    scanf("%d", &n);
    for(int i = 1; i <= n; i++){
        cin >> stu[i].nm >> stu[i].id >> stu[i].s;
    }

    sort(stu + 1, stu + 1 + n, cmp);

    cout << stu[n].nm << " " << stu[n].id << "\n";
    cout << stu[1].nm << " " << stu[1].id;
    return 0;
}
```

🚩 常见错误提醒

错误类型	具体表现
数组越界	忘记结构体数组从 1 开始存储
排序区间错误	<code>sort</code> 的区间传参不正确，如写成 <code>stu, stu+n</code> 而非 <code>stu+1, stu+1+n</code>
输入输出格式错误	忘记换行或多输出空格
compare函数错误	没写 <code>return x.s < y.s</code> 或逻辑反了

✅ 总结归纳

- 熟练掌握结构体的定义与数组存储
- 掌握 `sort` 函数的基本用法，特别是传参细节和自定义排序函数的写法
- 学会抽象封装数据对象（如学生），这是更复杂数据建模的基础

🧠 思维拓展

- 如果要查找成绩前 k 高或前 k 低的学生，可以结合 `priority_queue` 或 `partial_sort` 实现。
- 当排序规则复杂（如多重条件），可扩展为多关键字排序。