

奖学金（算法）

某小学最近得到了一笔赞助，打算拿出其中一部分为学习成绩优秀的前 5 名学生发奖学金。

期末，每个学生都有 3 门课的成绩:语文、数学、英语。

先按总分从高到低排序，如果两个同学总分相同，再按语文成绩从高到低排序，如果两个同学总分和语文成绩都相同，那么规定学号小的同学排在前面，这样，每个学生的排序是唯一确定的。

任务：先根据输入的 3 门课的成绩计算总分，然后按上述规则排序，最后按排名顺序输出前五名学生的学号和总分。

注意，在前 5 名同学中，每个人的奖学金都不相同，因此，你必须严格按上述规则排序。

例如，在某个正确答案中，如果前两行的输出数据(每行输出两个数:学号、总分) 是：

```
7 279
5 279
```

这两行数据的含义是:总分最高的两个同学的学号依次是 7 号、5 号。

这两名同学的总分都是 279 (总分等于输入的语文、数学、英语三科成绩之和), 但学号为 7 的学生语文成绩更高一些。

如果你的前两名的输出数据是:

```
5 279
7 279
```

则按输出错误处理。

输入格式

输入文件包含 $n + 1$ 行:

第 1 行为一个正整数 n ，表示该校参加评选的学生人数。

第 2 到 $n + 1$ 行，每行有 3 个用空格隔开的数字，每个数字都在 0 到 100 之间，第 j 行的 3 个数字依次表示学号为 $j - 1$ 的学生的语文、数学、英语的成绩。

每个学生的学号按照输入顺序编号为 $1 \sim n$ (恰好是输入数据的行号减 1)。

所给的数据都是正确的，不必检验。

输出格式

输出文件共有 5 行，每行是两个用空格隔开的正整数，依次表示前 5 名学生的学号和总分。

数据范围

$6 \leq n \leq 300$

输入样例：

```
6
90 67 80
87 66 91
78 89 91
88 99 77
67 89 64
78 89 98
```

输出样例：

```
6 265
4 264
3 258
2 244
1 237
```

y总代码：

```

//重载小于号
#include <iostream>
#include <cstring>
#include <algorithm>

using namespace std;

const int N = 310;

int n;
struct Person
{
    int id, sum, a, b, c;
    bool operator< (const Person& t) const
    {
        if (sum != t.sum) return sum > t.sum;
        if (a != t.a) return a > t.a;
        return id < t.id;
    }
}q[N];

int main()
{
    cin >> n;
    for (int i = 1; i <= n; i ++ )
    {
        int a, b, c;
        cin >> a >> b >> c;
        q[i] = {i, a + b + c, a, b, c};
    }

    sort(q + 1, q + n + 1);

    for (int i = 1; i <= 5; i ++ )
        cout << q[i].id << ' ' << q[i].sum << endl;

    return 0;
}

```

作者: yxc

链接: <https://www.acwing.com/activity/content/code/content/708261/>

来源: AcWing

著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。

```

//自定义比较函数
#include <iostream>
#include <cstring>
#include <algorithm>

using namespace std;

const int N = 310;

int n;
struct Person
{
    int id, sum, a, b, c;

```

```

}q[N];

bool cmp(Person& a, Person& b)
{
    if (a.sum != b.sum) return a.sum > b.sum;
    if (a.a != b.a) return a.a > b.a;
    return a.id < b.id;
}

int main()
{
    cin >> n;
    for (int i = 1; i <= n; i ++ )
    {
        int a, b, c;
        cin >> a >> b >> c;
        q[i] = {i, a + b + c, a, b, c};
    }

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

    for (int i = 1; i <= 5; i ++ )
        cout << q[i].id << ' ' << q[i].sum << endl;

    return 0;
}

```

作者: yxc

链接: <https://www.acwing.com/activity/content/code/content/708261/>

来源: AcWing

著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。