

明明的随机数

NOIP2006年普及组第一题。

问题描述

明明想在学校中请一些同学一起做一项问卷调查，为了实验的客观性，他先用计算机生成了N个1到1000之间的随机整数（N≤100），对于其中重复的数字，只保留一个，把其余相同的数去掉，不同的数对应着不同的学生的学号。然后再把这些数从小到大排序，按照排好的顺序去找同学做调查。请你协助明明完成“去重”与“排序”的工作。

输入格式

- 有2行，第1行为1个正整数，表示所生成的随机数的个数：N；
- 第2行有N个用空格隔开的正整数，为所产生的随机数。

输出格式

- 第1行为1个正整数M，表示不相同的随机数的个数。
- 第2行为M个用空格隔开的正整数，为从小到大排好序的不相同的随机数。

输入输出样例1

输入	输出
10 20 40 32 67 40 20 89 300 400 15	8 15 20 32 40 67 89 300 400

参考代码

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

int n,num[105];

int main(){
    cin >> n;
    for(int i=0;i<n;i++){
        cin >> num[i];
    }
    sort(num,num+n);
    int flag = num[0],cnt=1;
    for(int i=1;i<n;i++){
        if(num[i]!=flag){
            flag=num[i];
            cnt ++;
        }
    }
    cout << cnt << endl;
    flag = num[0];
    cout << flag;
    for(int i=1;i<n;i++){
        if(num[i]!=flag){
            flag=num[i];
            cout << " " << flag;
        }
    }
    cout << endl;
```

```
    return 0;
}
```

奖学金

NOIP2007年普及组第一题。

问题描述

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，不能得分。

输入格式

- 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~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

- 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。

输入输出样例1

输入	输出
6	6 265
90 67 80	4 264
87 66 91	3 258
78 89 91	2 244
88 99 77	1 237
67 89 64	
78 89 98	

输入输出样例2

输入	输出
8	8 265

80 89 89	2 264
88 98 78	6 264
90 67 80	1 258
87 66 91	5 258
78 89 91	
88 99 77	
67 89 64	
78 89 98	

数据说明

- 50%的数据满足:各学生的总成绩各不相同;
- 100%的数据满足: $6 \leq n \leq 300$ 。

参考代码

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

struct Student{
    int no;
    int chinese;
    int math;
    int english;
    int total;
};

Student s[305];
int n;

bool comp(Student a,Student b){
    if(a.total>b.total)
        return true;
    else if(a.total == b.total)
        if(a.chinese>b.chinese)
            return true;
        else if(a.chinese == b.chinese)
            if(a.no<b.no)
                return true;
    return false;
}

int main(){
    cin >> n;
    for(int i=1;i<=n;i++){
        s[i].no=i;
        cin >> s[i].chinese >> s[i].math >> s[i].english;
        s[i].total = s[i].chinese+s[i].math+s[i].english;
    }

    sort(s+1,s+n+1,comp);

    for(int i=1;i<=5;i++){
        cout << s[i].no << " " << s[i].total << endl;
    }

    return 0;
}
```

分数线划定

问题描述

世博会志愿者的选拔工作正在A市如火如荼的进行。为了选拔最合适的人才，A市对所有报名的选手进行了笔试，笔试分数达到面试分数线的选手方可进入面试。面试分数线根据计划录取人数的150%划定，即如果计划录取m名志愿者，则面试分数线为排名第m*150%（向下取整）名的选手的分数，而最终进入面试的选手为笔试成绩不低于面试分数线的所有选手。

现在就请你编写程序划定面试分数线，并输出所有进入面试的选手的报名号和笔试成绩。

输入格式

- 第一行，两个整数n, m ($5 \leq n \leq 5000$, $3 \leq m \leq n$)，中间用一个空格隔开，其中n表示报名参加笔试的选手总数，m表示计划录取的志愿者人数。输入数据保证m*150%向下取整后小于等于n。
- 第二行到第n+1行，每行包括两个整数，中间用一个空格隔开，分别是选手的报名号k ($1000 \leq k \leq 9999$) 和该选手的笔试成绩s ($1 \leq s \leq 100$)。数据保证选手的报名号各不相同。

输出格式

- 第一行，有两个整数，用一个空格隔开，第一个整数表示面试分数线；第二个整数为进入面试的选手的实际人数。
- 从第二行开始，每行包含两个整数，中间用一个空格隔开，分别表示进入面试的选手的报名号和笔试成绩，按照笔试成绩从高到低输出，如果成绩相同，则按报名号由小到大的顺序输出。

输入输出样例1

输入	输出
6 3	88 5
1000 90	1005 95
3239 88	2390 95
2390 95	1000 90
7231 84	1001 88
1005 95	3239 88
1001 88	

样例说明：m*150% = 3*150% = 4.5，向下取整后为4。保证4个人进入面试的分数线为88，但因为88有重分，所以所有成绩大于等于88的选手都可以进入面试，故最终有5个人进入面试。

参考代码

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

int n,m,line;

struct Candidate{
    int id;
    int score;
};

Candidate candi[5005];

bool comp(Candidate a,Candidate b){
    if(a.score>b.score)
        return true;
    else if(a.score == b.score)
        if(a.id < b.id)
```

```

        return true;
    return false;
}

int main(){
    cin >> n >> m;
    for(int i=0;i<n;i++){
        cin >> candi[i].id >> candi[i].score;
    }
    sort(candi,candi+n,comp);
    int p = m*1.5;
    int cnt = 1;
    for(int i=0;i<n;i++){
        if(cnt<p){
            cnt++;
        }else{
            line = candi[i].score;
            break;
        }
    }
    cnt = 0;
    for(int i=0;i<n;i++){
        if(candi[i].score>=line){
            cnt++;
        }else{
            break;
        }
    }
    cout << line << " " << cnt << endl;
    for(int i=0;i<cnt;i++){
        cout << candi[i].id << " " << candi[i].score << endl;
    }

    return 0;
}

```