

# 1997 年全国青少年信息学（计算机）奥林匹克竞赛试题

## 第一试

### 竞赛排名

某市组织了一次中学生科技全能竞赛，每个选手要参加数学、物理、化学、天文、地理、生物、计算机和英语共八项竞赛，最后综合八项竞赛的成绩排出总名次。选手编号依次为：1,2...N (N 为参赛总人数)。

设  $x_{ij}$  分别表示编号为 i 的选手第 j 项竞赛的成绩 ( $1 \leq i \leq N, 1 \leq j \leq 8$ )。其它指标如下：

● 第 j 项竞赛的平均分  $avg_j = \frac{1}{N} \sum_{i=1}^N x_{ij}, (1 \leq j \leq 8)$

● 选手 i 的总分  $sumx_i = \sum_{j=1}^8 x_{ij}, (1 \leq i \leq N)$

● 选手 i 第 j 项竞赛的位置分

$$y_{ij} = \begin{cases} 0, & (\sum_{i=1}^N |x_{ij} - avg_j| = 0) \\ \frac{x_{ij} - avg_j}{\frac{1}{N} \sum_{i=1}^N |x_{ij} - avg_j|}, & (\sum_{i=1}^N |x_{ij} - avg_j| \neq 0) \end{cases} \quad (1 \leq i \leq N, 1 \leq j \leq 8)$$

● 选手 i 的总位置分  $sumy_i = \sum_{k=1}^3 y_{ik} + 0.8 \sum_{k=4}^8 y_{ik}, (1 \leq i \leq N)$

排名规则如下：

1 总位置分高的选手名次在前：

2 若两个或两个以上的选手总位置分相同，则总分高的选手名次在前：

3 若两个或两个以上的选手总位置分和总分均相同，则编号在前的选手名次在前。

请你为竞赛组委会编一程序，计算本次全能竞赛的总排名情况。

### 输入输出

输入文件为 INPUT.TXT。文件的第一行为参赛总人数 N ( $1 \leq N \leq 1000$ )，从第二行到第 N 行依次为编号为 1 到编号为 N 的选手的成绩，每行有 8 个 0~100 之间的整数，代表该选手的 8 项竞赛成绩  $x_{i1}, x_{i2}, \dots, x_{i8}$ 。同一行相邻两个数之间用一个空格符隔开。

输出文件为 OUTPUT.TXT。文件有 N 行，从第 1 行到第 N 行依次为排名第 1 的选手的编号，排名第 2 的选手的编号，...，排名第 N 的选手的编号。

## 样例

### INPUT.TXT

```
3
72 82 73 68 95 86 82 90
72 90 50 60 80 70 65 80
72 82 73 68 95 86 82 90
```

### OUTPUT.TXT

```
1
3
2
```

## 最优乘车

H 城是一个旅游胜地，每年都有成千上万的人前来观光。为方便游客，巴士公司在各个旅游景点及宾馆，饭店等地都设置了巴士站并开通了一些单程巴士线路。每条单程巴士线路从某个巴士站出发，依次途经若干个巴士站，最终到达终点巴士站。

一名旅客最近到 H 城旅游，他很想去看 S 公园游玩，但如果从他所在的饭店没有一路巴士可以直接到达 S 公园，则他可能要先乘某一路巴士坐几站，再下来换乘同一站台的另一路巴士，这样换乘几次后到达 S 公园。

现在用整数  $1, 2, \dots, N$  给 H 城的所有的巴士站编号，约定这名旅客所在饭店的巴士站编号为 1，S 公园巴士站的编号为 N。

写一个程序，帮助这名旅客寻找一个最优乘车方案，使他在从饭店乘车到 S 公园的过程中换车的次数最少。

## 输入输出

输入文件是 INPUT.TXT。文件的第一行有两个数字 M 和 N ( $1 \leq M \leq 100$   $1 < N \leq 500$ )，表

示开通了 M 条单程巴士线路，总共有 N 个车站。从第二行到第 M 行依次给出了第 1 条到第 M 条巴士线路的信息。其中第 i+1 行给出的是第 i 条巴士线路的信息，从左至右按运行顺序依次给出了该线路上的所有站号相邻两个站号之间用一个空格隔开。

输出文件是 OUTPUT.TXT，文件只有一行。如果无法乘巴士从饭店到达 S 公园，则输出 "NO"，否则输出你的程序所找到的最少换车次数，换车次数为 0 表示不需换车即可到达。

## 样例

### INPUT.TXT

```
3 7
6 7
4 7 3 6
2 1 3 5
```

### OUTPUT.TXT

## 文件匹配

在计算机的日常操作中，经常需要对当前目录下所有文件中的一部分文件进行操作。例如，将当前目录下的所有文本文件复制至另一个目录下：将当前目录下所有以 a 打头的文件删除；等等。

很多操作系统都采用正则表达式来实现文件匹配功能。一种简单的正则表达式由英文字母（区分大小写）。数字及通配符“\*”和“?”组成，“?”代表任意一个字符，“\*”则可以代表零个或任意多个字符。

例如：

a\*b 可以匹配 acb (\*代表 c)

可以匹配 aabb (\*代表 ab)

可以匹配 asdsfddb (\*代表 sdsfdd)

可以匹配 ab (\*不代表任何字符)

a\*b 不可以匹配 ac (缺少最右边的字母 b)

不可以匹配 bb (缺少最左边的字母 a)

不可以匹配 abbc (最右边的字母不是 b)

a?b 可以匹配 acb (?代表 c)

不可以匹配 ab (缺少中间的一个字符)

不可以匹配 accb (?只能代表一个字符)

现要对某目录下的部分文件进行操作。写一个程序,寻找一个正则表达式，使其能匹配的待操作文件最多，但不能匹配任何不进行操作的文件。注意你所找到的最优正则表达式的长度应当是最短的。如果有多个长度最短的最优正则表达式，则其中任意一个都是允许的。

### 输入输出

输入文件是 INPUT.TXT.文件由 N ( $1 \leq N \leq 250$ ) 行组成。每行给出了一个文件名（由英文字母和数字组成：英文字符要区分大小写，文件名长度不超过 8 个字符），其后是一个空格符和一个字符（“+”或“-”）。“+”表示要对该行给出的文件进行操作，“-”表示不进行操作。

输出文件是 OUTPUT.TXT，文件由两行组成。第一行是一个整数，给出了你的程序所找到的最优正则表达式所能匹配的文件数目。在第二行给出你的程序所找到的最优正则表达式。

### 样例

#### INPUT.TXT

EXCHANGE +

EXT +

HARDWARE +

MOUSE -

NETWORK -

#### OUTPUT.TXT

2

E +