

# 2023 NOI 春季测试

时间：2023 年 3 月 4 日 08:30 ~ 13:00

题目名称	涂色游戏	幂次	圣诞树	密码锁
题目类型	传统型	传统型	传统型	传统型
目录	paint	power	tree	lock
可执行文件名	paint	power	tree	lock
输入文件名	paint.in	power.in	tree.in	lock.in
输出文件名	paint.out	power.out	tree.out	lock.out
每个测试点时限	1.0 秒	1.0 秒	1.0 秒	2.5 秒
内存限制	512 MiB	1024 MiB	1024 MiB	512 MiB
测试点数目	20	20	20	20
测试点是否等分	是	是	是	是

提交源程序文件名

对于 C++ 语言	paint.cpp	power.cpp	tree.cpp	lock.cpp
-----------	-----------	-----------	----------	----------

编译选项

对于 C++ 语言	-O2 -std=c++14 -static
-----------	------------------------

注意事项（请仔细阅读）

1. 文件名（程序名和输入输出文件名）必须使用英文小写。
2. C/C++ 中函数 main() 的返回值类型必须是 int，程序正常结束时的返回值必须是 0。
3. 提交的程序代码文件的放置位置请参考各省的具体要求。
4. 因违反以上三点而出现的错误或问题，申诉时一律不予受理。
5. 若无特殊说明，结果的比较方式为全文比较（过滤行末空格及文末回车）。
6. 选手提交的程序源文件必须不大于 100KB。
7. 程序可使用的栈空间内存限制与题目的内存限制一致。
8. 全国统一评测时采用的机器配置为：Intel(R) Core(TM) i7-8700K CPU @3.70GHz，内存 32GB。上述时限以此配置为准。
9. 只提供 Linux 格式附加样例文件。
10. 评测在当前最新公布的 NOI Linux 下进行，各语言的编译器版本以此为准。

# 涂色游戏 (paint)

## 【题目描述】

有一天,小 D 在刷朋友圈时看到了一段游戏视频。

这个游戏的名字叫涂色游戏,视频中的游戏界面是一个  $n$  行  $m$  列的网格,初始时每一个格子都是白色(用数字 0 表示)。其中每一行的左侧、每一列的上方都有一把带颜色的刷子。玩家点击某个刷子后,这个刷子会将其右侧(或下方)的一整行(或一整列)涂上同一种颜色,该行(或该列)格子原有的颜色都会被覆盖成新涂上的颜色。

下图展示的情况可以通过先将第一列涂成红色,然后将第一行涂成蓝色得到,若此时选择将第三列涂成绿色,则图中绿色方框中的格子都会变成绿色。

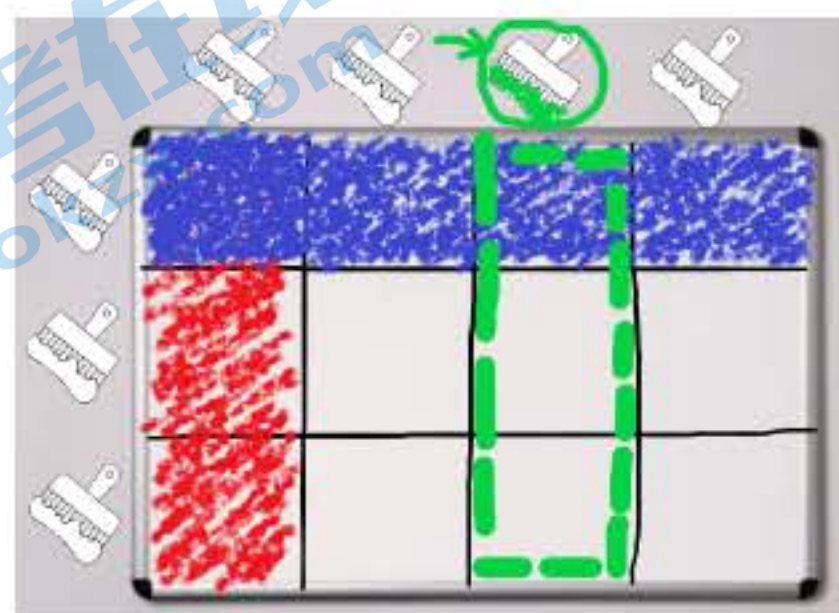


图 1: 涂色示例

小 D 想用他自己编写的程序来进行视频中的游戏。在编程的过程中,小 D 在涂色逻辑的实现上却遇到了一些困难,于是他向你求助,希望你能帮他完成实现涂色逻辑部分的代码。

首先,小 D 会给你网格的行数和列数  $n, m$ , 然后给出  $q$  次操作,每次操作用三个整数  $opt_i, x_i, c_i$  表示:

- 如果  $opt_i = 0$ , 那么这次操作会将第  $x_i$  行涂成颜色  $c_i$ 。
- 如果  $opt_i = 1$ , 那么这次操作会将第  $x_i$  列涂成颜色  $c_i$ 。

在所有涂色操作结束以后,你需要输出网格中每个位置的颜色是什么。

## 【输入格式】

从文件 `paint.in` 中读入数据。

本题有多组测试数据。

第一行包含一个正整数  $T$ , 表示数据组数。

接下来一共  $T$  组数据, 每组数据格式如下:

第一行包含三个整数  $n, m, q$ , 分别表示涂色板的行数、列数, 以及小 D 进行涂色操作的次数。

接下来  $q$  行, 每行包含三个整数  $opt_i, x_i, c_i$ , 表示一次操作。

### 【输出格式】

输出到文件 `paint.out` 中。

对于每组数据，输出  $n$  行，每行  $m$  个由单个空格隔开的整数。

其中第  $i$  行第  $j$  个整数表示涂色完成后网格中第  $i$  行第  $j$  列的方格是什么颜色。

### 【样例 1 输入】

```
1 2
2 5 5 9
3 1 5 1
4 0 4 0
5 1 4 1
6 0 3 0
7 1 3 1
8 0 2 0
9 1 2 1
10 0 1 0
11 1 1 1
12 3 3 3
13 0 1 2
14 0 3 1
15 1 1 3
```

### 【样例 1 输出】

```
1 1 0 0 0 0
2 1 1 0 0 0
3 1 1 1 0 0
4 1 1 1 1 0
5 1 1 1 1 1
6 3 2 2
7 3 0 0
8 3 1 1
```

### 【样例 1 解释】

注意当一个格子没有被涂色时，其颜色为白色，用数字 0 表示。

## 【样例 2】

见选手目录下的 `paint/paint2.in` 与 `paint/paint2.ans`。

## 【数据范围】

对于所有数据，保证：

- $1 \leq T \leq 10$ ,  $1 \leq n, m \leq 10^5$ ,  $0 \leq q \leq 10^5$ ,  $0 \leq c_i \leq 10^9$ 。
- 若  $opt_i = 0$ , 则  $1 \leq x_i \leq n$ ; 若  $opt_i = 1$ , 则  $1 \leq x_i \leq m$ 。
- 单个测试点中所有数据的  $n \cdot m$  的总和不超过  $10^6$ ,  $q$  的总和不超过  $10^6$ 。

测试点	$n \leq$	$m \leq$	$q \leq$	性质 A	性质 B		
1	1	10	0	√	√		
2			1				
3			20				
4			$10^5$	$10^5$		×	
5							×
6							
7	10	10	20	√	√		
8	50	50	100				
9						×	
10	1000	1000	2000	×	√		
11					×		
12						×	
13							×
14							
15	$10^5$	$10^5$	$10^5$	√	×		
16						×	
17							×
18				×			
19						×	
20	×						

特殊性质 A：保证测试点中所有的  $q \cdot \max(n, m)$  之和不超  $10^7$ 。

特殊性质 B：保证  $opt_i = 1$ 。

## 【提示】

数据千万条，清空第一条。多测不清空，爆零两行泪。

关注北京高考在线官方微信：[北京高考资讯\(微信号:bjgkzx\)](#)，获取更多试题资料及排名分析信息。

# 幂次 (power)

## 【题目描述】

小  $\Omega$  在小学数学课上学到了“幂次”的概念： $\forall a, b \in \mathbb{N}^+$ ，定义  $a^b$  为  $b$  个  $a$  相乘。她很好奇有多少正整数可以被表示为上述  $a^b$  的形式？由于所有正整数  $m \in \mathbb{N}^+$  总是可以被表示为  $m^1$  的形式，因此她要求上述的表示中，必须有  $b \geq k$ ，其中  $k$  是她事先选取好的一个正整数。

因此她想知道在 1 到  $n$  中，有多少正整数  $x$  可以被表示为  $x = a^b$  的形式，其中  $a, b$  都是正整数，且  $b \geq k$ ？

## 【输入格式】

从文件 `power.in` 中读入数据。

第一行包含两个正整数  $n, k$ ，意义如上所述。

## 【输出格式】

输出到文件 `power.out` 中。

输出一行包含一个非负整数表示对应的答案。

## 【样例 1 输入】

```
1 99 1
```

## 【样例 1 输出】

```
1 99
```

## 【样例 1 解释】

由于所有正整数  $x \in [1, 99]$  总可以表示为  $x = x^1$  的形式，因此答案是 99。

## 【样例 2 输入】

```
1 99 3
```

### 【样例 2 输出】

1 7

### 【样例 2 解释】

以下是全部 7 组符合题意的正整数及对应的一种合法的表示方法。

$$1 = 1^3, 8 = 2^3, 16 = 2^4, 27 = 3^3, 32 = 2^5, 64 = 4^3, 81 = 3^4$$

注意某些正整数可能有多种合法的表示方法，例如 64 还可以表示为  $64 = 2^6$ 。但根据题意，同一个数的不同的合法表示方法只会被计入一次。

### 【样例 3 输入】

1 99 2

### 【样例 3 输出】

1 12

### 【样例 3 解释】

以下是全部 12 组符合题意的正整数及对应的一种合法的表示方法。

$$1 = 1^2, 4 = 2^2, 8 = 2^3, 9 = 3^2, 16 = 4^2, 25 = 5^2$$

$$27 = 3^3, 32 = 2^5, 36 = 6^2, 49 = 7^2, 64 = 8^2, 81 = 9^2$$

### 【样例 4】

见选手目录下的 `power/power4.in` 与 `power/power4.ans`。

### 【样例 5】

见选手目录下的 `power/power5.in` 与 `power/power5.ans`。

### 【样例 6】

见选手目录下的 `power/power6.in` 与 `power/power6.ans`。

### 【数据范围】

对于所有数据，保证  $1 \leq n \leq 10^{18}$ ， $1 \leq k \leq 100$ 。

测试点编号	$n \leq$	$k$
1	$10^2$	$= 1$
2		$\geq 2$
3	$10^4$	$\geq 3$
4		$\geq 2$
5	$10^6$	$\geq 3$
6		$\geq 2$
7	$10^8$	$\geq 3$
8		$\geq 2$
9	$10^{10}$	$\geq 3$
10		$\geq 2$
11	$10^{12}$	$\geq 3$
12		$\geq 2$
13	$10^{14}$	$\geq 3$
14		$\geq 2$
15	$10^{16}$	$\geq 3$
16		$\geq 2$
17	$10^{18}$	$\geq 3$
18		$\geq 2$
19		
20		

# 圣诞树 (tree)

## 【题目描述】

众所周知，3202 年的圣诞节快要到了，因此小  $\Omega$  买了一颗圣诞树和一根挂满了彩灯的电线，并打算把这根电线缠绕在圣诞树上。

圣诞树可以视作一个二维平面上有  $n$  个顶点的凸多边形。这  $n$  个顶点可以用于固定电线，且按顺时针顺序依次编号为  $1, \dots, n$ 。其中第  $i$  个顶点的坐标为  $(x_i, y_i)$ ，记其中  $y$  坐标最大的顶点的编号为  $k$ （若有多个满足条件的顶点，则取编号最小的）。

下图左侧展示了一棵圣诞树的轮廓，其中  $y$  坐标最大的顶点的编号为  $k = 5$ 。



图 2: 一棵圣诞树及一种可能的挂电线的方案

小  $\Omega$  希望用挂满了彩灯的电线装饰这颗圣诞树。出于美观性考虑，她希望这个电线经过所有顶点恰好一次；为了连接电源，这根电线需要从  $(x_k, y_k)$  出发。形式化地，她需要决定一个  $1, \dots, n$  的排列  $p_1, \dots, p_n$ ，满足  $p_1 = k$ ，随后这根电线从  $(x_{p_1}, y_{p_1})$  出发，依次经过  $(x_{p_2}, y_{p_2}), \dots, (x_{p_n}, y_{p_n})$ 。此时，电线长度为  $\sum_{i=1}^{n-1} d((x_{p_i}, y_{p_i}), (x_{p_{i+1}}, y_{p_{i+1}}))$ 。

• 其中  $d$  为平面上的欧几里得距离，即  $d((x, y), (x', y')) = \sqrt{(x - x')^2 + (y - y')^2}$ 。

上图右侧展示了一种可能的方案，此时对应的排列为  $5, 4, 8, 6, 3, 9, 1, 7, 2$ 。

为了节省成本，她希望你能在所有可能的方案中，给出一种使电线长度最短的方案。如果使电线长度最短的方案不唯一，你只需要求出其中任意一种。

考虑到浮点数产生的误差，你输出的方案与最优方案的线段长度的相对误差或绝对误差不超过  $10^{-10}$  时即认为答案正确。

## 【输入格式】

从文件 `tree.in` 中读入数据。



第一行包含一个正整数  $n$ ，表示圣诞树的顶点数。

接下来  $n$  行，其中第  $i$  行包含两个精确到小数点后 9 位的实数  $x_i, y_i$  表示编号为  $i$  的顶点的坐标。

数据保证这  $n$  个点两两不同，并且依次连接  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$  将形成一个凸多边形。

### 【输出格式】

输出到文件 `tree.out` 中。

输出一行包含  $n$  个由单个空格隔开的正整数  $p_1, p_2, \dots, p_n$ ，表示一个  $1, \dots, n$  的排列，满足  $p_1 = k$ ，且电线的长度  $\sum_{i=1}^{n-1} d((x_{p_i}, y_{p_i}), (x_{p_{i+1}}, y_{p_{i+1}}))$  在所有可能的方案中最短。如果这样的方案不唯一，请输出其中任意一种方案。

### 【样例 1 输入】

```
1 3
2 0.000000000 0.000000000
3 3.000000000 0.000000000
4 1.000000000 1.000000000
```

### 【样例 1 输出】

```
1 3 1 2
```

### 【样例 1 解释】

这一样例中只有下图所示的两种方案，对应排列分别为 3, 1, 2 或 3, 2, 1，电线长度分别为  $3 + \sqrt{2}$  和  $3 + \sqrt{5}$ ，而  $3 + \sqrt{2} < 3 + \sqrt{5}$ 。

因此答案对应的排列为 3, 1, 2。

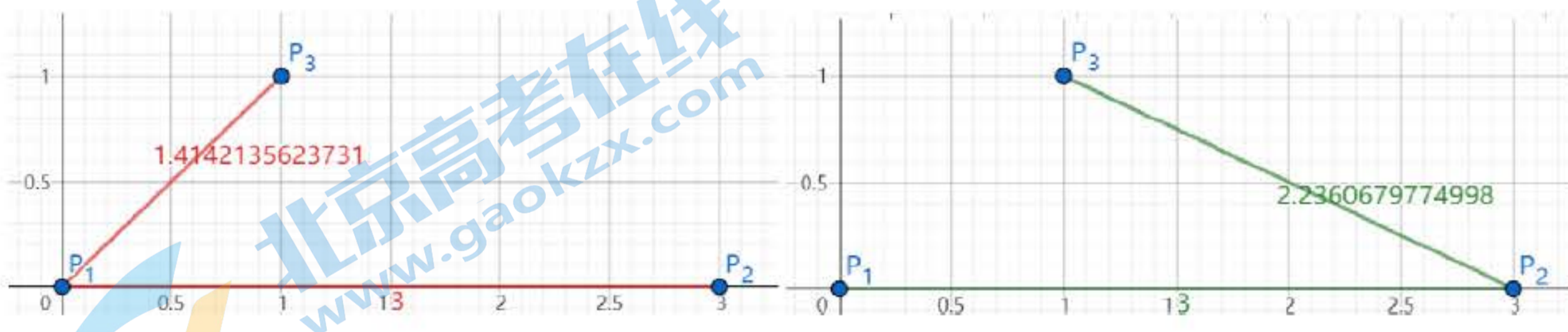


图 3: 样例 1 的全部两种可能的方案

### 【样例 2】

见选手目录下的 *tree/tree2.in* 与 *tree/tree2.ans*。

### 【样例 3】

见选手目录下的 *tree/tree3.in* 与 *tree/tree3.ans*。

### 【样例 4】

见选手目录下的 *tree/tree4.in* 与 *tree/tree4.ans*。

该样例数据满足特殊性质 A。

### 【样例 5】

见选手目录下的 *tree/tree5.in* 与 *tree/tree5.ans*。

该样例数据满足特殊性质 B。

### 【样例 6】

见选手目录下的 *tree/tree6.in* 与 *tree/tree6.ans*。

### 【数据范围】

对于所有数据，保证  $3 \leq n \leq 1000$ ； $|x_i|, |y_i| \leq 10^7$ 。

测试点编号	$n \leq$	特殊性质
1,2	4	无
3,4,5,6	9	
7,8,9,10,11,12	18	
13,14	$10^3$	A
15,16		B
17,18,19,20		无

特殊性质 A：保证存在正整数  $m \geq n$ ，使得输入的  $n$  个顶点对应正  $m$  边形中连续的一段顶点。

特殊性质 B：保证  $x_1 < x_2 < \dots < x_n$ ，且  $y_1 > y_2 > \dots > y_n$ 。

# 密码锁 (lock)

## 【题目描述】

寒假过后，小 I 回到学校，发现自己忘记了自行车锁的密码，于是请你帮忙。

小 I 自行车上的密码锁有  $n$  个拨圈，每个拨圈有  $k$  ( $k \leq 4$ ) 格。密码锁上的每一格都包含一个正整数，其中第  $j$  个拨圈的第  $i$  格上的正整数为  $a_{i,j}$ 。



图 4: 一个锁的例子，其中  $k = n = 3$ ，每列表示一个拨圈，拨圈的格子从上往下编号。

你可以对每个拨圈拨若干次（也可以不拨），每拨一次拨圈，它的格子就会进行一次轮换。形式化地，拨第  $j$  个拨圈一次，则会让第  $j$  个拨圈上第  $i$  格的数字移动到第  $((i \bmod k) + 1)$  格，其他拨圈不动。

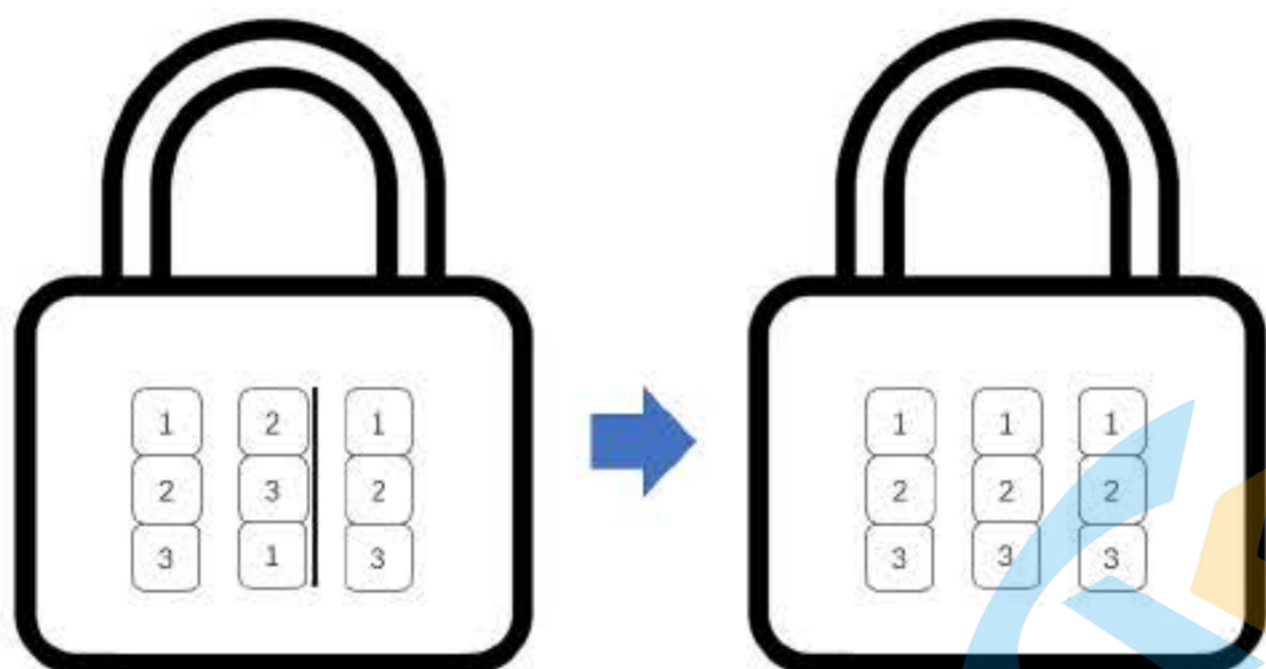


图 5: 一个拨动拨圈的例子，对左侧的锁拨一次第二个拨圈得到右侧的锁。

为了方便记忆，小 I 设定密码时要求同一行上的数字尽可能靠近。

形式化地，对于  $1 \leq i \leq k$ ，定义密码锁第  $i$  行的松散度为

$$c(i) = \max_{j=1}^n a_{i,j} - \min_{j=1}^n a_{i,j}$$

同时定义整个密码锁的松散度为

$$C = \max_{1 \leq i \leq k} c(i)$$

因为能开锁的状态满足  $C$  尽可能小，因此小 I 希望你找出最小的  $C$  值。

## 【输入格式】

从文件 `lock.in` 中读入数据。

本题有多组测试数据，题目保证一个测试点中所有测试数据的  $k$  相同。

第一行包含两个正整数  $T, k$ ，分别表示测试数据组数和密码锁拨圈上的格数。

接下来一共  $T$  组数据，每组数据格式如下：

第一行包含一个正整数  $n$ ，表示拨圈数。

接下来  $k$  行，每行包含  $n$  个正整数，其中第  $i$  行第  $j$  个整数  $a_{i,j}$  表示密码锁第  $j$  个拨圈上第  $i$  格对应的数字。

注意输入的矩阵中每一列对应一个拨圈，而非每一行对应一个拨圈。

## 【输出格式】

输出到文件 `lock.out` 中。

对于每组数据，输出一行包含一个整数，表示所有方案中  $C$  的最小值。

## 【样例 1 输入】

```
1 2 3
2 3
3 1 2 1
4 2 3 2
5 3 1 3
6 2
7 1 2
8 2 1
9 1 2
```

## 【样例 1 输出】

```
1 0
2 1
```

## 【样例 1 解释】

第一组样例对应题目描述中的例子。

在拨第二个拨圈一次后，每个拨圈都是  $\{1, 2, 3\}$ ，此时松散度为 0。

容易证明无论如何松散度都不可能小于 0，因此输出 0。

以下四个样例分别对应  $k = 1, 2, 3, 4$  的情况，且样例中  $n$  的取值有一定梯度。

### 【样例 2】

见选手目录下的 `lock/lock2.in` 与 `lock/lock2.ans`。

### 【样例 3】

见选手目录下的 `lock/lock3.in` 与 `lock/lock3.ans`。

### 【样例 4】

见选手目录下的 `lock/lock4.in` 与 `lock/lock4.ans`。

### 【样例 5】

见选手目录下的 `lock/lock5.in` 与 `lock/lock5.ans`。

### 【数据范围】

设  $\sum n$  为一个测试点中所有测试数据的  $n$  的和。

对于所有数据，保证  $1 \leq T$ ， $1 \leq k \leq 4$ ， $1 \leq a_{i,j} \leq 3 \times 10^4$ 。

本题分为两类测试点。

第一类测试点共有十二个，保证  $k \leq 3$ ， $n \leq 5 \times 10^4$ ， $\sum n \leq 1.5 \times 10^5$ 。

测试点编号	$n \leq$	$\sum n \leq$	$k =$
1	20	100	1
2	$5 \times 10^4$	$1.5 \times 10^5$	
3	20	100	2
4	100	1000	
5	2000	$10^4$	
6	$5 \times 10^4$	$1.5 \times 10^5$	
7	10	50	3
8	50	500	
9	300	3000	
10	3000	$2 \times 10^4$	
11	$3 \times 10^4$	$1.2 \times 10^5$	
12	$5 \times 10^4$	$1.5 \times 10^5$	

第二类测试点共有八个，保证  $k = 4$ ， $n \leq 10^4$ ， $\sum n \leq 3 \times 10^4$ 。

测试点编号	$n \leq$	$\sum n \leq$	$k =$
13	10	50	4
14	50	500	
15	200	2000	
16	500	4000	
17	2500	$10^4$	
18	5000	$2 \times 10^4$	
19	$10^4$	$3 \times 10^4$	
20			

### 【后记】

你花了九牛二虎之力算出  $C$  的值之后，小 I 却告诉你他已经找开锁师傅用锤子暴力破解了。在你的百般劝说下，小 I 承诺以后锁车不用有大于等于一万个拨圈的密码锁。

## 关于我们

北京高考在线创办于 2014 年，隶属于北京太星网络科技有限公司，是北京地区极具影响力的中学升学服务平台。主营业务涵盖：北京新高考、高中生涯规划、志愿填报、强基计划、综合评价招生和学科竞赛等。

北京高考在线旗下拥有网站门户、微信公众平台等全媒体矩阵生态平台。平台活跃用户 40W+，网站年度流量数千万量级。用户群体立足于北京，辐射全国 31 省市。

北京高考在线平台一直秉承 “精益求精、专业严谨” 的建设理念，不断探索 “K12 教育+互联网+大数据” 的运营模式，尝试基于大数据理论为广大中学和家长提供新鲜的高考资讯、专业的高考政策解读、科学的升学规划等，为广大高校、中学和教科研单位提供 “衔接和桥梁纽带” 作用。

平台自创办以来，为众多重点大学发现和推荐优秀生源，和北京近百所中学达成合作关系，累计举办线上线下升学公益讲座数百场，帮助数十万考生顺利通过考入理想大学，在家长、考生、中学和社会各界具有广泛的口碑影响力

未来，北京高考在线平台将立足于北京新高考改革，基于对北京高考政策研究及北京高校资源优势，更好的服务全国高中家长和学生。



微信搜一搜

北京高考资讯