

# 2007 年安徽省安联杯信息学奥林匹克竞赛

## AHOI 2007

### 第一试

比赛时间: 2007 年 4 月 21 日 8:00 至 11:00

题目名称	密码箱	宫殿大门	灯阵
可执行文件名	box.exe	Door.exe	light.exe
输入文件名	box.in	Door.in	light.in
输出文件名	box.out	Door.out	light.out
试题类型	传统型	传统型	传统型
满分	100	100	100
是否有部分分	否	否	否
时限	1 秒	1 秒	2 秒

#### 注意事项

1. 务必看清题目, 严格按照所要求的格式输入、输出。
2. 在调试程序时请先使用题目中的示例数据, 然后再自行设计多组测试数据进行调试。
3. 测试有严格的时间限制, 请尽可能优化算法。
4. 命名规则:
  - (1) 每题都规定了该题的英文名称。
  - (2) 程序文件和数据文件的主文件名都是该题的英文名字。
  - (3) 程序文件扩展名采用语言环境的默认扩展名。
  - (4) 数据文件都是文本文件, 输入和输出文件的扩展名分别是.in 和.out。
5. 程序应从输入文件读取数据, 并严格地按照规定的输出格式将结果输出到输出文件中。输入数据文件和输出数据文件都与程序在同一个目录中, 由于程

序所在目录是不确定的，因此不允许在程序中含有盘符信息和任何形式的路径信息。

6. 选手在竞赛结束时应在硬盘指定位置建立以参赛号命名的文件夹，并将所完成各题的源程序文件和编译所产生的可执行文件(即扩展名为.exe 的文件)拷贝到该文件夹中。

## 题目

### 1. 密码箱 ( Box )

在一次偶然的情况下，小可可得到了一个密码箱，听说里面藏着一份古代流传下来的藏宝图，只要能破解密码就能打开箱子，而箱子背面刻着的古代图标，就是对密码的提示。经过艰苦的破译，小可可发现，这些图标表示一个数以及这个数与密码的关系。假设这个数是  $n$ ，密码为  $x$ ，那么可以得到如下表述：

**密码  $x$  大于等于 0，且小于  $n$ ，而  $x$  的平方除以  $n$ ，得到的余数为 1。**

小可可知道满足上述条件的  $x$  可能不止一个，所以一定要把所有满足条件的  $x$  计算出来，密码肯定就在其中。计算的过程是很艰苦的，你能否编写一个程序来帮助小可可呢？（题中  $x$ ， $n$  均为正整数）

输入：输入文件只有一行，且只有一个数字  $n$  ( $1 \leq n \leq 2,000,000,000$ )。

输出：你的程序需要找到所有满足前面所描述条件的  $x$ ，如果不存在这样的  $x$ ，

你的程序只需输出一行“None”（引号不输出），否则请按照从小到大的顺序

输出这些  $x$ ，每行一个数。

样例：

输入：

12

输出：

1

5

7

11

## 2. 宫殿大门 ( Door )

密码箱终于被打开了，里面真的有一张古代的藏宝图，这是一个古代君主留下来的宝藏，藏在一个至今没有人发现的地下宫殿中，要取得宝藏必须破解重重机关。小可可毫无畏惧，作为他的好朋友，你们就这样开始了寻宝旅程。经过千辛万苦，你们来到了地下宫殿的入口，遇到了第一个麻烦。宫殿的大门怎么都无法打开。小可可仔细观察以后发现，大门的左半扇门上有一些数字，像一个  $n*n$  的矩阵般排列，右半扇门对应的位置上也有同样的  $n*n$  数字排列，不过数字均为 0，这些 0 都可以通过机关变成其他的数字，甚至是负数，但是排列方式不会变。小可可仔细研究了藏宝图，发现要打开这扇门必须要将右半扇门上的数字按照一定的规律进行改变。若假设左边  $n*n$  矩阵为 A，改变后的右边  $n*n$  矩阵为 B，这个规律就是：

它们的乘积 C 矩阵： $C_{ij} = \sum_{k=1}^n a_{i,k} * b_{k,j}$ ，且  $C=A*B=E_n$

其中  $E_n$  为： $E_{ij}=0$  (若  $i$  不等于  $j$ )， $E_{ij}=1$  (若  $i$  等于  $j$ )。即，B 为 A 的逆矩阵。

你能帮助小可可打开这扇门吗？

输入：输入文件的第一行为一个整数  $n$  ( $1 \leq n \leq 200$ )。

以下  $n$  行，每行  $n$  个数，按照行主序给出 A 的每个元素，即：

$A_{1,1}, A_{1,2}, \dots, A_{1,n}$ ，

$A_{2,1}, A_{2,2}, \dots, A_{2,n}$ ，

$\dots$

$A_{n,1}, A_{n,2}, \dots, A_{n,n}$ .

输出：输出文件为  $n$  行，每行  $n$  个整数，且在  $-2,000,000,000$  与  $2,000,000,000$  之间，每两个相邻的数用一个空格隔开，即按照行主序表示的矩阵  $B$ 。

输入数据保证这个矩阵  $B$  存在，且  $B$  的每个元素为一个整数，并且在  $-2,000,000,000$  与  $2,000,000,000$  之间。

特别提示：由于测试采取文件比对方式，所以这里约定，如果结果中某个数为 0，则应输出 0，而不是 -0。

样例：

输入：

```
3
1 -1 1
0 1 2
1 0 4
```

输出：

```
4 4 -3
2 3 -2
-1 -1 1
```

### 3. 灯阵 ( *Light* )

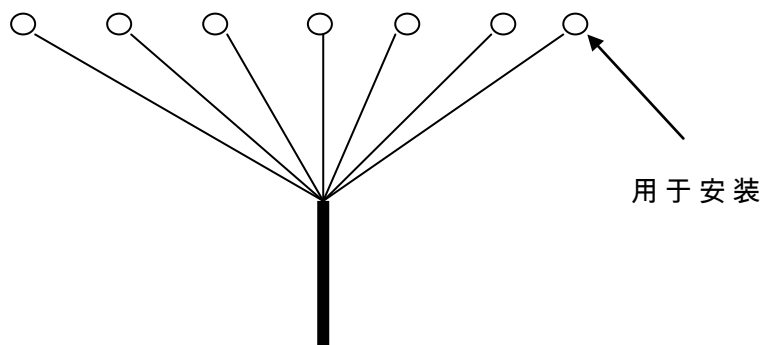
宫殿的大门打开了，里面漆黑一片，藏宝图上记载，必须要点亮灯阵，宫殿内的各种陷阱才会关闭，才能进入宫殿深处，否则每一步都有极大的危险。这个灯阵是由很多盏油灯组成，它们排成长长的一行。这些油灯经过巧妙的设计，有很多种型号，每种型号都有专用的点灯工具。假设每盏灯都有一个类型编号  $p$  ( $1 \leq p \leq 2,000,000,000$ )。那么点灯工具也有不同的编号  $s$  ( $1 \leq s \leq 2,000,000,000$ )。每盏灯只能被与自己编号相同的点灯工具点燃，例如

1 号类型的灯只能被编号为 1 的点灯工具点燃。灯阵旁边就有古代的点灯器，它是一个类似树杈的东西，后端一个粗把手，前端有一些分叉。只不过前端的分叉可以任意增加和减少，每个分叉末端都必须正好安装一种点灯工具，这样就可以同时点燃数个灯，但是点灯工具一旦安装上就无法去掉，所以分叉的个数与点灯工具的安装都要经过精细的计算。经过现场仔细的观察和对藏宝图的研究，小可可了解到灯的分布情况与点燃的方式为：

- 1、所有灯按顺序排成一行，位置从左到右分别为  $1, 2, 3, \dots, n$ ，称为 1 号位、2 号位等等，每两条相邻的灯的间隔是相等的，可以有重复型号的灯。
- 2、点灯器分叉的末端在一条直线上，上面必须挂  $m$  个点灯工具，每两个相邻的点灯工具的间隔相等，且等于每两条相邻的灯的间隔；
- 3、从最左边的灯开始，把点灯器上最左边的点灯工具与最左边的灯对齐，分叉末端点灯工具的排列方向与灯排列的方向相同，且必须保证所有点灯工具的编号与那个要点燃的灯的编号相等；然后就可以点灯了。灯一旦点燃就不会熄灭。
- 4、点灯的过程就是从最左边开始，不断向右走（从不往回走，也不掉转点灯器的方向），寻找下一个点灯的位置。必须仍然保证所有点灯工具的编号与要点燃的灯的编号相等，而且每个点灯工具下都必须有灯；
- 5、一定要保证所有的灯都被点燃（当然，可能有一些灯被点了多次）。只有这样，灯阵才算被真正点亮，也才能进行下一步探宝。

灯阵旁边有足够多的各种型号的点灯工具。如果点灯器前端的分叉数量大于等于 1 且小于等于  $n$ ，那么需要多少个分叉能够保证成功点亮灯阵呢？小可可

想麻烦你来帮他计算所有可能的情况,他会根据你的计算最终确定合适的点燃灯阵的方案。



点灯器示意图

输入: 输入文件中第一行有一个整数  $n$  ( $1 \leq n \leq 500,000$ ), 表示灯阵中灯的数目。

下面有  $n$  行每行有一个数字, 第  $i$  个数字表示从左数第  $i$  号位灯的类型编号  $p$  ( $1 \leq p \leq 2,000,000,000$ )。

输出: 输出所有满足要求的分叉的数目。每行输出一个数, 要求从小到大输出。

样例:

输入:

3  
2  
2  
2

输出:

1  
2  
3