

2025 华南师范大学“卓越教育杯”算法与程序设计竞赛（新生赛）

迫害小灯小纵队

2025 年 12 月 14 日



Problems

A	gugugaga
B	小橘的魔法宝石
C	图片压缩
D	Nuit Sans La Soleil
E	construction is 2 hard 4 me
F	$>v<$
G	小橘的糖果游戏
H	SCNU LOGO
I	升级宝可梦
J	扫雪

Problem A. gugugaga

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

小古是位刚入学的新生，怀揣着对 XCPC 的向往，在认真学习了一段时间后，她非常想检验一下自己的水平，于是她非常自信地找到了她的学长，让她的学长出题考考她，并且和学长打赌一定能写出来。学长思索过后，出了一道有点意思的题：

- 求出两个整数的 PCPD — Product of Common Prime Divisor，公素约数的乘积。

下面给出一个示例：

- 给定 $a = 60, b = 70$ ， a 的质因子有 $(2, 3, 5)$ ， b 的质因子有 $(2, 5, 7)$ 。
- 那么 $\text{PCPD}(a, b) = 2 \cdot 5 = 10$ 。
- 如果 a, b 没有公素约数，那么 $\text{PCPD}(a, b) = 0$ 。

小古如临大敌，但为了赢下和学长的赌约，她向现在正在看题的你发起了求助。



Input

每个测试文件由多个测试用例组成。第一行包含一个正整数 T ($1 \leq T \leq 2 \cdot 10^5$) 表示测试用例数。

对于每个测试用例，输入一行两个正整数 a, b ($2 \leq a, b \leq 2 \cdot 10^6$)。

Output

对于每个测试用例，输出一行一个整数表示 $\text{PCPD}(a, b)$ 。

Example

standard input	standard output
3	10
60 70	0
12 175	70
498050 88480	

Problem B. 小橘的魔法宝石

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

小橘有 n 颗魔法宝石，每颗宝石的魔力值为 a_i 。任意两颗宝石的**共同魔力**的值为 $a_i \& a_j^*$ 。小橘现在想知道，把**所有**可能的共同魔力全部加起来的值是多少？

换句话说，你需要计算：

$$\sum_{i=1}^{n-1} \sum_{j=i+1}^n (a_i \& a_j)$$

Input

每个测试文件由多个测试用例组成。第一行包含一个正整数 T ($1 \leq T \leq 100$) 表示测试用例数。

对于每个测试用例，第一行包含一个正整数 n ($1 \leq n \leq 2 \times 10^5$)，表示小橘的魔法宝石数。

第二行包含 n 个整数 $a_1, a_2, \dots, a_n$ ($0 \leq a_i \leq 10^8$)，表示每颗宝石的魔力值。

数据保证所有测试用例 n 的总和不超过 4×10^5 。

Output

对于每个测试用例，输出一个非负整数，表示所有可能的共同魔力值的和。

Example

standard input	standard output
2	3
3	21
1 2 3	
4	
3 5 6 7	

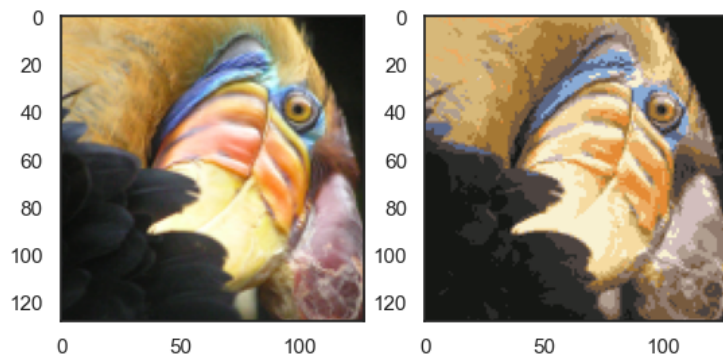
*公式中的 $\&$ 表示二进制数的按位与运算。

Problem C. 图片压缩

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

在计算机里，一张图片可以看成是一个由成千上万个微小的、带颜色的小方块组成的，这些小方块就是像素。它是图像中最小的基本单位。每一个像素点都对应着一组特定的 **RGB**（红、绿、蓝）数值。这组数值通常由三个数字组成，例如 (255, 0, 0)。每个数字代表了该颜色成分的强度或亮度。数值范围通常是 0 到 255，0 表示没有这种颜色，255 表示这种颜色的最大亮度。

图像压缩可以采用一种名为 *k-means* 聚类的机器学习算法，这种算法的思想是：将 **RGB** 张成的三维空间（欧几里得空间）中距离较近的某块聚簇，统一用一个颜色（一个特定的 RGB）表示。这样子可以大大减少要表示的 RGB 数量，但是难免地会有信息的 loss（损失）。



左图为压缩前的图，右图为压缩后的图

在了解上述知识后，你决定对某一张包含 $n \times m$ 个像素点的小图片进行压缩，使用一种颜色去表示这张小图片，你需要最小化 loss。

loss 定义：假设图中的原始像素点为： $(x_1, y_1, z_1), (x_2, y_2, z_2), \dots, (x_{n \times m}, y_{n \times m}, z_{n \times m})$ ，你选择的颜色为 (x, y, z) ，则 $\text{loss} = \sum_{i=1}^{n \times m} ((x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2)$ 。

Input

第一行两个正整数 n, m ($1 \leq n, m \leq 10^5, n \times m \leq 10^5$)，表示图片的尺寸。

接下来 n 行，每行有 m 个用字符串表示的 RGB 格式的像素点，每个像素点格式为 `xxx,xxx,xxx`。其中 `xxx` 是介于 0 ~ 255 之间的一个整数（不足三位将添加前导零补齐）。

即：用长度固定为 11 的一个字符串表示一个像素点的 RGB。

Output

输出一行一个整数，表示最小的 loss。

Example

standard input	standard output
2 2 000,000,000 000,255,000 255,000,000 255,255,000	130052

Note

样例中，最优的一种选择是 (127, 127, 000)。

Problem D. Nuit Sans La Soleil

Input file: standard input
 Output file: standard output
 Time limit: 1 second
 Memory limit: 256 megabytes

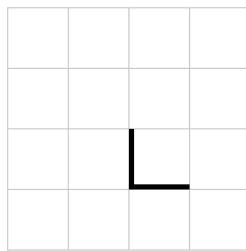
"客人，您的拿铁。"你醒来了，正趴在咖啡厅的一张桌子上，一名扎着两条马尾的白发少女给你递过来一杯咖啡。咖啡厅呈一个 L 形状，窗外像贴图一般诡异的一片空白。据少女所说，你是不久前凭空出现在这的，而这家咖啡厅名叫 *La Lune*，是和她一起被转移到这个空间的，到这里已经有两三天了。

你走到门口，门外全白的地面又带有无数整齐的灰线，延伸到视野尽头。少女跟上来，这段时间她一个人查清楚了这个世界中的一些规则，包括不知从何而来的纸条上写着的离开此处的方法。这个无人世界似乎是一张巨大的网格图，每个网格上都能凭意志放置或移除一家 *La Lune*。少女说着，指向远处，顺着她的手指方向，你看到原本空无一物的位置多了一家同样 L 形布局的咖啡厅，咖啡厅的外边缘恰好契合地面的网格线。"每家 *La Lune* 会刚好占用一个网格上，并且……"地面开始震动，"每隔一段时间，它们都会绕自己所在的网格中心旋转。"你勉强保持不摔倒。

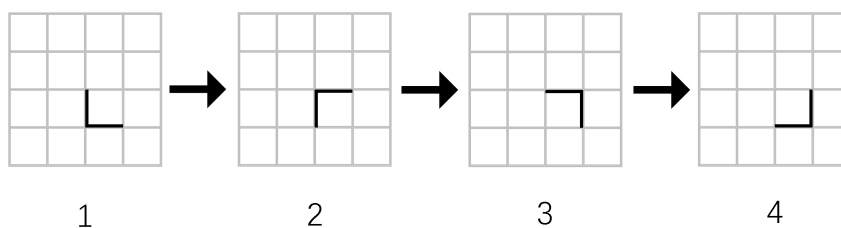
而纸条上写的离开的方法是：建立更多的餐厅吧，直到 k 总是包围你。纸条背面还列了大量的 k 等于多少的字样，差不多十几二十行。即使需要更多信息，少女也摇头道只知道这些。和朋友分开令她感到不安，她脑袋上两只猫耳稍稍耷拉下来，"巧克力……"少女呢喃。从新海市的异变到现在为止，这次似乎是少女第一次变的孤身一人。那是什么地方？你不知道，但你理解了纸条上的谜语，剩下的只是怎么建立餐厅了。

人类，上面的话都不重要喵 ----- pwp 参上 ~

这是一个 L 形的图案：



每过一个时刻，这个图案就会绕自己所在的单元格的中心顺时针旋转 90 度。



将 L 形图案顺时针旋转 0, 90, 180, 270 度得到的图案分别标为 1, 2, 3, 4，而空位就标号为 0 喵。

现在你可以在无限大的网格图上，任意放置四种方向的图案。

构造一个初始（第 0 时刻）的放置方案，使得图上存在一个位置 (x,y) ，在初始及接下来任一时刻，总是被包围在一个大小为 k 格（忽略图案自身粗细占据的面积）的封闭区域内。

Input

输入一行一个整数 k ($1 \leq k \leq 1000$)。

Output

如果不存在符合题意的方案，输出一行一个 -1 。否则：

第一行输出 4 个整数 n,m,x,y ($1 \leq n,m \leq 1000, 1 \leq x \leq n, 1 \leq y \leq m$)，表示你需要使用的网格图大小，以及符合题意的位置 (x,y) 。

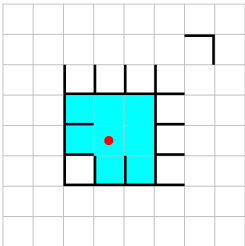
接下来 n 行每行 m 个数字，输出一个仅包含数字 0 到 4 的矩阵表示你的答案。同一行的 m 个数字间不加空格。

Example

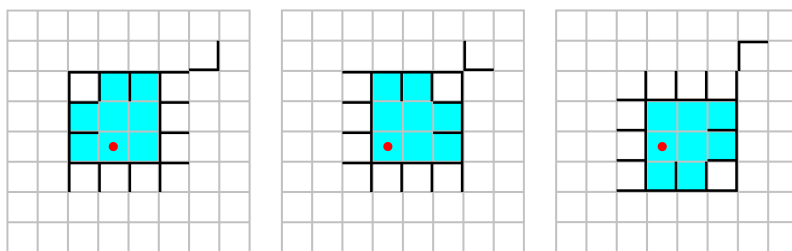
standard input	standard output
1	4 5 2 3 44444 44444 44444 44444
8	8 8 5 4 00000000 00000030 00111100 00100100 00100100 00111100 00000000 00000000

Note

对于样例 2，题目中给出方案的初始情况（第 0 时刻）如下：



在接下来的第 1, 2, 3 时刻，如下图依次从左到右。能看到图中的位置 (5, 4) 总是位于一个大小为 8 格的封闭区域（淡蓝色区域）内。



Problem E. construction is 2 hard 4 me

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

给定一个长度为 n 且严格单调递增的数列 $\{a_n\}$ 。
你可以将 $\{a_n\}$ 中的至多 k 个数删除，得到一个长度为 m 的数列 $\{b_m\}$ 。
最后，你将获得 $\min_{1 \leq i < m} (b_{i+1} - b_i)$ 的得分。
特别地，当 $m < 2$ 时，得分为 $+\infty$ 。
你需要给出一个删数方案，以最大化你的得分。
如果有多组合法方案，输出其中任意一组即可。

Input

每个测试文件由多个测试用例组成。第一行包含一个正整数 T ($1 \leq T \leq 100$) 表示测试用例数。
对于每个测试用例，第一行输入两个正整数 n, k ($1 \leq k \leq n \leq 2 \times 10^5$)。
接下来一行有 n 个正整数，表示数列 $\{a_n\}$ ($1 \leq a_i \leq 10^9, a_i < a_{i+1}$)。
数据保证所有测试用例 n 的总和不超过 2×10^5 。

Output

对于每个测试用例，第一行输出一个非负整数 d ($0 \leq d \leq k$)，表示 $\{b_m\}$ 由 $\{a_n\}$ 中删除 d 个数得到。
接下来一行有 d 个互不相同的正整数，表示删去的 d 个数原先在 $\{a_n\}$ 中的编号。

Example

standard input	standard output
2	1
5 1	3
2 4 5 6 8	3
4 3	2 3 4
2 7 9 10	

Problem F. $\geq v <$

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes



这是一道交互题。

欢迎光临魔法游戏大厅，我是交互式游戏机 1007 号，请输入文本.....

游戏机屏幕上有一个长度 ≤ 8 的非空字符串，下面称其为明码。

保证明码中仅含 $>$ 与 $<$ 两种字符。

你可以输入形如 UvO 的指令，其中 U 与 O 是互异且仅含 $>$ 与 $<$ 两种字符的非空字符串。

之后游戏机将执行你输入的指令 -----

具体来说，它会重复地在明码中寻找最靠前的 U 子串，并将其替换为 O ，直至满足下面两个条件之一：

- 明码中不存在 U 子串。
- 再次进行替换会使明码长度 ≥ 500 。

每当你输入的指令执行结束时，若明码满足下面两个条件之一，你就在本局游戏中获胜：

- 明码中 $>$ 的数量恰好为 249。
- 明码中 $<$ 的数量恰好为 249。

游戏机的设计者觉得这还太过简单，于是又添加了下面的限制：

- 每当你输入一条指令后，若你未获胜，游戏机会自行输入一条指令，之后你才能再次输入指令。

- 当游戏机输入 24 次指令后，若你尚未获胜，你就在本局游戏中落败。
- 当你第 i 次输入形如 UvO 的指令时，要求 U, O 的长度均不超过 i ，下面称其为**渐进约束**。
- 幸运的是，你可以看见游戏机输入的指令，并且游戏机输入指令时也会受到渐进约束。

你需要连续地游玩 T 局上述的游戏，并在 T 局游戏中全部获胜。

Input

第一行包含一个正整数 T ($1 \leq T \leq 512$)，表示游戏局数。

每局游戏的具体描述如下。

Interaction Protocol

对于每局游戏，最初会给出一个非空字符串，表示明码。

保证明码长度 ≤ 8 ，且仅包含 $>$ 与 $<$ 两种字符。

随后你可以输入若干次指令。

若你需要输入指令，请按照下面的格式输出新的一行：

- UvO ----- 要求 U, O 为互异且仅含 $>$ 与 $<$ 的非空字符串，表示输入的指令。

每当你输入指令后，游戏机会依次执行以下步骤：

- 检查你输入的指令是否合法。若是则继续执行下面的步骤；否则输出一行 -1 并结束交互。
- 执行你输入的指令。
- 检查你是否获胜。若是则输出一行 $!$ 并结束该局游戏；否则继续执行下面的步骤。
- 游戏机输出一行合法指令，并对明码执行该指令。
- 检查自身是否已输出 24 次指令。若是则输出一行 -1 并结束交互；否则等待你下一次输入指令。

若你在该局游戏中获胜，并且该局游戏不是最后一局游戏，则开始下一局游戏。

每次输出指令后，请输出换行并刷新缓冲区[†]；否则可能得到 `Idleness limit exceeded` 的评测结果。

若你在交互过程中读取到了 -1 ，这意味着你的程序给出了不合法的输出，或你在某局游戏中落败。

此时你应立刻终止程序，以取得 `Wrong Answer` 的评测结果；否则可能因在关闭的流中进行读入而得到除 `Accepted` 之外的任一评测结果。

请注意：该问题的交互器是自适应的。

这意味着，游戏机可以根据游戏情况来构造指令，以阻碍你通过本题。

[†]本场比赛支持的各语言分别可以通过如下函数刷新缓冲区：

- C++ 语言： `fflush(stdout)` 或 `cout.flush()`。
- C 语言： `fflush(stdout)`。
- Java 语言： `System.out.flush()`。
- Python 语言： `stdout.flush()`。

Example

standard input	standard output
1	
>>>>>>>>	
	>v<
<v>	
	>>v<>
<>v><	
	><v<<>
<>v><<	
	><v<<<>
<>v><<<	
	><<<<v<
!	

Note

下面是样例解释：

最初明码为从左到右 8 个 >。

你输入指令 >v<，明码变为 8 个 <。

游戏机输入指令 <v>，明码变为 8 个 >。

你输入指令 >>v<>，明码变为 7 个 < 与 1 个 >。

游戏机输入指令 <>v><，明码变为 1 个 > 与 7 个 <。

你输入指令 ><v<<>，明码变为 14 个 < 与 1 个 >。

游戏机输入指令 <>v><<，明码变为 1 个 > 与 28 个 <。

你输入指令 ><v<<<>，明码变为 84 个 < 与 1 个 >。

游戏机输入指令 <>v><<<，明码变为 1 个 > 与 252 个 <。

你输入指令 ><<<<v<，明码变为 249 个 <，你在本局游戏中获胜。

请注意：游戏机可不一定会像样例中这样乖巧地配合你。

Problem G. 小橘的糖果游戏

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

ice~ 来看小橘找到了什么！（小橘从身后掏出一个大大的糖果罐子）
这是一个装满了 n 颗糖果的罐子。小橘决定和主人玩一个游戏来分配这些糖果。
作为一只尊老爱幼爱主人的猫娘，小橘决定让主人占点便宜！
所以——小橘先手！（喵哈哈！）
游戏规则如下：

- 1. 罐子里总共有 n 颗糖果。
- 2. 小橘和 ice 轮流从罐子中取糖。
- 3. 小橘永远先手。
- 4. 每个人在自己的回合，必须选择吃掉 1 颗、3 颗或 4 颗糖果。
- 5. 如果轮到某一方时，他/她无法按照规则取糖（即无法取出 1、3 或 4 颗或者罐子空了），那么该方失败。

假设小橘和 ice 都绝顶聪明（尤其是 ice!），并且都会采取最优策略，请问最后是谁会赢下这场游戏？

Input

每个测试文件由多个测试用例组成。第一行包含一个正整数 T ($1 \leq T \leq 10^5$) 表示测试用例数。
对于每个测试用例，输入一行一个非负整数 n ($0 \leq n \leq 10^9$)，表示糖果的总数。

Output

对于每个测试用例，输出一行一个字符串：
如果小橘获胜，请输出 Orange；
如果 ice 获胜，请输出 ice。

Example

standard input	standard output
4	ice
2	Orange
4	ice
7	Orange
8	

Problem H. SCNU LOGO

Input file: standard input
 Output file: standard output
 Time limit: 1 second
 Memory limit: 256 megabytes

Welcome to SCNU 2025 Freshman Programming Contest!

作为一名新生, 你收到了一个用 * 和 . 组成的 SCNU LOGO。原始的 LOGO 为一个 5×36 的二维字符矩阵:

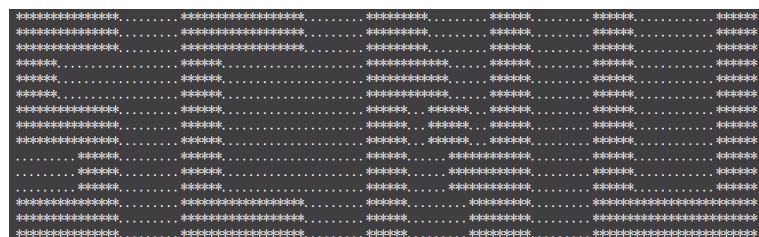
```
*****. . .*****. . .*****. . .**. . .**. . .**
**. . .**. . .**. . .**. . .**. . .**
*****. . .**. . .**. . .**. . .**
. . .**. . .**. . .**. . .**
*****. . .*****. . .**. . .*****
```

现在, hank 想请你把这个 LOGO 放大到原来的 k ($1 \leq k \leq 5$) 倍, 你需要输出放大后的 LOGO。

Tips: 放大可以看成把原先的一个像素变成一个 $k \times k$ 的像素。比如下面是 $k = 2$ 的结果:

```
*****. . .*****. . .*****. . .*****. . .*****
*****. . .*****. . .*****. . .*****. . .*****
****. . .****. . .*****. . .****. . .****. . .*****
****. . .****. . .*****. . .****. . .****. . .*****
*****. . .****. . .*****. . .****. . .****. . .*****
*****. . .****. . .*****. . .****. . .****. . .*****
. . .****. . .****. . .*****. . .*****. . .****. . .*****
. . .****. . .****. . .*****. . .*****. . .****. . .*****
*****. . .*****. . .****. . .*****. . .*****
*****. . .*****. . .****. . .*****. . .*****
```

$k = 3$ 的结果如下:



Input

输入一个正整数 k ($1 \leq k \leq 5$), 表示放大倍数。

Output

输出放大 k 倍后的 LOGO。

Example

standard input	standard output
1	*****.....*****.....**.....**.....** **.....**.....*****.....**.....**.....** *****.....**.....**.....**.....**.....** ...**.....**.....**.....*****.....**.....** *****.....*****.....**.....**.....*****.....

Problem 1. 升级宝可梦

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

冰冰妙妙屋的 Minecraft 方块宝可梦服务器盛大开服啦！

作为服务器的第一批尊贵玩家，ice、sanjuu、dusk_cloud、luckybird、rain 和 buzhi_ 几乎在同一时间登录了游戏。

宝可梦可以使用经验水晶来升级。当一个宝可梦从 $k - 1$ 级升到 k 级的时候，需要消耗 $k - 1$ 颗水晶。

例如：

- 从 1 级升到 2 级，需要消耗 1 颗水晶。
- 从 2 级升到 3 级，需要消耗 2 颗水晶。
- ... 以此类推。

现在，这六位玩家分别抓到了一只 1 级的宝可梦，他们想要将自己的宝可梦一口气喂到各自指定的等级！

请你帮他们计算，每位玩家分别需要多少颗经验水晶呢？

Input

输入一行六个整数，分别代表六位玩家想要将自己宝可梦喂养到的等级 $a_1, a_2, \dots, a_6$ ($1 \leq a_i \leq 10^9$)。

Output

输出一行六个整数，分别代表六位玩家需要的经验水晶个数。

Example

standard input	standard output
10 8 3 4 6 2	45 28 3 6 15 1

Problem J. 扫雪

Input file: standard input
Output file: standard output
Time limit: 4 second
Memory limit: 1024 megabytes

北国风光，万里雪飘。大雪刚刚过去，Galaxy 抵达哈尔滨的一个小镇。
镇上早已被大雪封路，唯一的一辆扫雪车正在工作，努力为小镇清理积雪。
该小镇共有 n 条双向车道，可以视为确定端点的 n 条线段。清雪作业需要遵循以下规则：

- 扫雪车在积雪路段上行驶时，将清除该路段的单向积雪；要清除对向车道的积雪，需要反向行驶。
- 扫雪车不能逆行，但可以在任意位置掉头到对向车道；两条或多条车道可能交汇于路口，扫雪车可以在路口驶向任意车道。
- 扫雪车需要从一个给定的起点出发，清扫小镇中所有车道的双向积雪。一条车道的积雪不必连续清除，但必须保证所有车道的所有位置都在某一时刻被扫雪车经过。
- 忽略扫雪车掉头所需的距离以及车道宽度。

为了尽快完成除雪、恢复通行，Galaxy 想要找到一个使扫雪车总行驶距离最短的方案。在此之前，他需要你计算这个最短行驶距离。

Input

输入的第一行包含一个整数 n ($1 \leq n \leq 10^4$)，表示双向车道的数量。
接下来 n 行，每行四个整数 S_x, S_y, T_x, T_y ($0 \leq S_x, S_y, T_x, T_y \leq 10^5$)，表示一条车道两个端点 S, T 的横纵坐标。
最后一行给出扫雪车的起始位置 (x, y) ($0 \leq x, y \leq 10^5$)，并保证该点是某条车道的一个端点。
即：存在某条车道满足 $(S_x, S_y) = (x, y)$ 或 $(T_x, T_y) = (x, y)$ 。
保证输入的所有路径相互连通，且任意两条线段的交至多只有一个点。

Output

输出一个实数 D ，表示扫雪车完成全部除雪所需的最短行驶距离。
如果你的输出与标准答案 D' 的绝对误差或相对误差不超过 10^{-6} ，即满足 $\frac{|D-D'|}{\max(1, |D'|)} \leq 10^{-6}$ ，则视为正确答案。

Example

standard input	standard output
2 0 0 2 2 4 0 0 4 0 0	16.9705627485