



2023 年第五届河南省 CCPC 大学生程序设计竞赛 正式赛

2023 年 5 月 7 日



题目概况

题号	题目名	时间限制	空间限制
A	小水獭游河南	1 s	256 MB
B	Art for Rest	1 s	256 MB
C	Toxel 与随机数生成器	1 s	256 MB
D	Toxel 与多彩的宝可梦世界	8 s	256 MB
E	矩阵游戏	2 s	256 MB
F	Art for Last	1 s	256 MB
G	Toxel 与字符画	1 s	256 MB
H	Travel Begins	1 s	256 MB
I	数正方形	1 s	256 MB
J	Mocha 沉迷电子游戏	2 s	256 MB
K	排列与质数	1 s	256 MB
L	猜数游戏	6 s	256 MB



Problem A. 小水獭游河南

输入文件: standard input
输出文件: standard output
时间限制: 1 second
空间限制: 256 megabytes

小水獭来到河南旅游, 它认为一个字符串 s 是 **HENAN** 的当且仅当存在两个非空字符串 a 和 b 满足如下三个条件:

- a 由小写字母组成, 且 a 中每种字母只出现了一次。
- b 由小写字母组成, 且 b 是回文串, 也就是说将 b 翻转后得到的字符串和 b 相同。
- 将 a 和 b 顺序拼接得到的字符串和 s 相同, 也就是说 $s = a + b$ 。

例如 **henan** 是 **HENAN** 的, 因为 **henan=he+nan**, 此外也可分解为 **hena+n**。但 **hhnan** 和 **ysmihoyocom** 不是 **HENAN** 的。

现在给定一个字符串, 请你帮助小水獭判断它是不是 **HENAN** 的。

输入格式

本题包含多组数据。

第一行包含一个整数 T ($1 \leq T \leq 10^3$), 表示数据组数。

对于每组数据:

一行包含一个由小写字母组成的字符串 s ($1 \leq |s| \leq 10^5$), 表示小水獭询问的字符串。

保证所有数据的 $\sum |s| \leq 10^5$ 。

输出格式

对于每组数据:

输出一行包含一个字符串。如果 s 是 **HENAN** 的, 输出 **HE**; 否则输出 **NaN**。

样例

standard input	standard output
3	HE
henan	NaN
hhnan	NaN
ysmihoyocom	

Problem B. Art for Rest

输入文件: standard input

输出文件: standard output

时间限制: 1 second

空间限制: 256 megabytes

给定一个长度为 n 的非负整数序列 a_1, a_2, \dots, a_n , 记作 A 。

对于正整数 k , 按照以下方式得到序列 A'_k :

- 将 A 划分为 $\lceil \frac{n}{k} \rceil$ 段, 第 i 段为 $a_{k(i-1)+1}, a_{k(i-1)+2}, \dots, a_{\min\{ki, n\}}$;
- 每一段升序排序后依次连接得到 A'_k 。

试求有多少个 k 满足 $1 \leq k \leq n$, 且对于任意 $1 \leq i < j \leq n$ 有 $A'_{k,i} \leq A'_{k,j}$ 。

输入格式

第一行包含一个正整数 n ($1 \leq n \leq 10^6$), 表示非负整数序列 A 的长度。

第二行包含 n 个非负整数 a_1, \dots, a_n ($0 \leq a_i \leq 10^9$), 表示给定的序列 A 。

输出格式

一行包含一个整数, 表示答案。

样例

standard input	standard output
13 1 1 4 5 1 4 1 9 1 9 8 1 0	1
4 114 514 1919 810	2
6 121 117 114 105 107 111	1

Problem C. Toxel 与随机数生成器

输入文件: standard input

输出文件: standard output

时间限制: 1 second

空间限制: 256 megabytes

题目背景: 知名游戏公司 *KaerfEmag* 在其最新游戏 *Telracs & Teloiv* 中生成随机数的算法有很大的漏洞, 生成的序列 “不够随机”, 严重影响玩家的游戏体验。作为 *KaerfEmag* 公司的项目负责人, *Toxel* 需要修复这一漏洞。

这里有一个简单的随机数生成器 XORSHIFT64:

```
uint64_t rng = seed; // uint64_t 一般可认为是 unsigned long long

uint64_t xorshift64(){
    rng ^= rng << 13;
    rng ^= rng >> 7;
    rng ^= rng << 17;
    return rng;
}
```

只要提供初始种子 *seed*, 不断调用 XORSHIFT64 函数, 就可以产生随机的 0/1 序列:

算法 1 正确的随机序列生成器

```
1:  $rng \leftarrow seed$ 
2: for  $i \leftarrow 1$  to  $10^6$  do
3:    $randomBit \leftarrow \text{XORSHIFT64}() \bmod 2$ 
4:   PRINT( $randomBit$ )
5: end for
```

不幸的是, *Telracs & Teloiv* 代码中的随机序列生成器是不正确的:

算法 2 错误的随机序列生成器

```
1:  $lengthList \leftarrow [length_1, length_2, \dots, length_m]$ 
2: for each  $length \in lengthList$  do
3:    $rng \leftarrow seed$  ▷ 错误地将  $rng$  重置为了初始种子  $seed$ 
4:   for  $i \leftarrow 1$  to  $length$  do
5:      $randomBit \leftarrow \text{XORSHIFT64}() \bmod 2$ 
6:     PRINT( $randomBit$ )
7:   end for
8: end for
```

这会导致序列随机性不够强, 每隔 $length_i$ 个位置, 序列就会开始重复。

现给出一段 0/1 序列, 请你判断生成它的随机序列生成器是正确的还是错误的。

输入格式

一行一个字符串 s ($|s| = 10^6$), 仅包含字符 0 或 1, 保证该字符串由题目所描述的两随机序列生成器之一生成。保证 $10^3 \leq length_i \leq 10^4$ 且 $\sum_{i=1}^m length_i = 10^6$ 。但是, 样例受篇幅所限, 不满足前述数据范围。样例不会出现在实际测试数据中。

输出格式

输出一行一个字符串。如果生成输入字符串的是正确的随机序列生成器, 则输出 **Yes**, 否则输出 **No** (大小写不敏感)。

样例

standard input
000101110010010000111000100000010001000001101111001010010110 000101001011011010111011100010101101110111000101101011010000 010001110000000010111010101010011001110010011011010101111001 001100100100100110001011101000101100010111100110100011000010 001101011001111101011010000110101010010000000111010101110110
standard output
Yes

standard input
000101110010010000111000100000010001000001101111001010010110 000101110010010000111000100000010001000001101111001010010110 000101110010010000111000100000010001000001101111001010010110 000101110010010000111000100000010001000001101111001010010110 000101110010010000111000100000010001000001101111001010010110
standard output
No

提示

在两个样例中, $seed$ 均为 227。第一个样例对应的随机数生成器是正确的, 因为中途 `rng` 没有重置过。第二个样例中, $lengthList = [60, 60, 60, 60]$ 。

在题面中, 样例受篇幅所限, 每 60 个字符为一行。但实际数据中行内不会出现换行符。

Problem D. Toxel 与多彩的宝可梦世界

输入文件: standard input

输出文件: standard output

时间限制: 8 seconds

空间限制: 256 megabytes

广阔的宝可梦世界是由 n 个城镇（编号为 $1, 2, \dots, n$ ）、 m 条道路组成的。每条道路连接两个城镇，这两个城镇均可通过这条道路走到对方。两个城镇间可能有多条道路连接。更特别的是，有的道路两端可能连接同一个城镇。换言之，宝可梦世界构成一个无向图，且不一定为简单无向图。

宝可梦世界中每条道路都有一个固定的颜色，颜色可能是 $1 \sim k$ 之间的整数。现在，Toxel 定义一个城镇集合是多彩的，当且仅当对于 k 种颜色中的每一种，集合中任意两个城镇可以只经过集合内的城镇及该颜色的道路到达对方。

Toxel 希望能够找到最大的多彩城镇集合，请你帮助它求解。

输入格式

本题包含多组数据。

第一行包含一个整数 T ($1 \leq T \leq 2 \times 10^5$)，表示数据组数。

对于每组数据：

第一行包含三个整数 k, n, m ($1 \leq k \leq 10^9, 1 \leq n \leq 500, 0 \leq m \leq 2 \times 10^5$)。

接下来 m 行，每行包含三个整数 g_i, u_i, v_i ($1 \leq g_i \leq k, 1 \leq u_i, v_i \leq n$)，表示第 i 条道路颜色为 g_i ，连接 u_i 和 v_i 。

保证所有数据的 $\sum m \leq 2 \times 10^5$ 。

输出格式

对于每组数据：

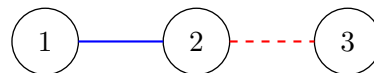
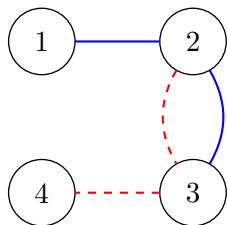
输出第一行包含一个整数 c ，表示最大的多彩城镇集合大小。第二行包含 c 个整数，表示集合中的所有城镇的编号。你可以以任意顺序输出集合中的城镇。如果有多种答案满足多彩城镇集合最大，你可以输出任意一种。

样例

standard input	standard output
2	2
2 4 4	2 3
1 1 2	1
1 2 3	1
2 2 3	
2 3 4	
2 3 2	
1 1 2	
2 2 3	

提示

样例中两组数据分别对应下方的图：



第一组数据中，2 和 3 之间包含全部两种颜色的边，因此显然在每种颜色下都可以互相到达， $\{2, 3\}$ 满足题目要求。

Problem E. 矩阵游戏

输入文件: standard input
输出文件: standard output
时间限制: 2 seconds
空间限制: 256 megabytes

Bellalabella 正在玩一个矩阵游戏。游戏中有一个 $n \times m$ 的矩阵，仅包含 01? 三种字符。 n 行从上到下依次用 $1, 2, \dots, n$ 编号， m 列从左到右依次用 $1, 2, \dots, m$ 编号。Bellalabella 可以将矩阵中不超过 x 个 ? 变为 1。Bellalabella 开始在矩阵的左上角 $(1, 1)$ ，并且只能向下或者向右走，最终在 (n, m) 停止，在移动的过程中每经过一个 1 得分会增加一分（包括起点和终点），经过其它字符则分数不变。开始分数为 0 分。

Bellalabella 在修改矩阵后，会以最优的策略从左上角移动到右下角。他想知道在最优的修改策略下，最多能获得多少分。

输入格式

本题包含多组数据。

第一行包含一个整数 T ($1 \leq T \leq 2.5 \times 10^4$)，表示数据组数。

对于每组数据：

第一行包含三个整数 n, m, x ($1 \leq n, m \leq 500, 0 \leq x \leq 10^3$)。

接下来 n 行，每行包含一个长度为 m 的字符串，保证只包含 01? 三种字符。

保证所有数据的 $\sum n \cdot m \leq 2.5 \times 10^5$ 。

输出格式

对于每组数据：

输出一行包含一个整数，表示替换不超过 x 个 ? 的前提下，最多能获得多少分。

样例

standard input	standard output
1 3 3 1 000 001 ?00	1
1 3 3 2 000 001 ??0	2

Problem F. Art for Last

输入文件: standard input
输出文件: standard output
时间限制: 1 second
空间限制: 256 megabytes

给定一个长度为 n 的非负整数序列 A_1, A_2, \dots, A_n , 其中 $n \geq 2$ 。

给定正整数 k , 满足 $2 \leq k \leq n$ 。要求从 A_1, A_2, \dots, A_n 中选择 k 项 $A_{p_1}, A_{p_2}, \dots, A_{p_k}$ ($1 \leq p_1 < p_2 < \dots < p_k \leq n$), 使得下式取得最小值:

$$\min_{1 \leq i < j \leq k} \{|A_{p_i} - A_{p_j}|\} \cdot \max_{1 \leq i < j \leq k} \{|A_{p_i} - A_{p_j}|\}$$

试求上式能够取得的最小值。

输入格式

第一行包含两个正整数 n, k ($2 \leq k \leq n \leq 5 \times 10^5$), 表示非负整数序列 A 的长度及选取的项数。

第二行包含 n 个非负整数 A_1, \dots, A_n ($0 \leq A_i \leq 10^9$), 表示给定的序列 A 。

输出格式

一行包含一个整数, 表示答案。

样例

standard input	standard output
13 7 1 1 4 5 1 4 1 9 1 9 8 1 0	0
4 2 114 514 1919 810	87616
6 3 121 117 114 105 107 111	12

提示

对于样例二, 最优解选择 A_2, A_4 , 答案为 $|514 - 810| \times |514 - 810| = 87616$;

对于样例三, 最优解选择 A_4, A_5, A_6 , 答案为 $|105 - 107| \times |105 - 111| = 12$ 。

Problem G. Toxel 与字符画

输入文件: standard input

输出文件: standard output

时间限制: 1 second

空间限制: 256 megabytes

一天, Toxel 找到了一个形如 x^y 的数学表达式, 其中 x 和 y 都是正整数。他希望通过“自我包含”的方式将这个表达式绘制成字符画。

具体来说, Toxel 首先将计算出 x^y 的结果等于 z 。若 $z \leq 10^{18}$, Toxel 将把 $x^y = z$ 绘制出来; 否则, 他将绘制 $x^y = \text{INF}$ 。Toxel 将在一个 $10 \times m$ 像素的画布上绘制表达式。非指数部分的数字, 以及 $=\text{INF}$ 这四个符号每个占据 7×7 像素的位置, 且其顶端与画布顶端距离为 2。指数部分的数字每个占据 5×5 像素的位置, 且其顶端与画布顶端距离为 1。每个数字或符号所包含的像素均填入本字符, 其余空白像素则填入 `.` 字符。数字和符号的具体形态请参见末尾的提示部分。任意两个符号之间应间隔恰好一列空白像素, 且左右两端也应留出恰一列空白像素。需要注意的是, 字符 `1` 也需要占满 7×7 (非上标) 或 5×5 (上标) 像素。请参阅样例以更好地理解题意。

Toxel 最近忙于给河南省赛命题, 因此没有时间自己来绘制字符画了。请你帮他完成这个任务。

输入格式

本题包含多组数据。

第一行包含一个整数 T ($1 \leq T \leq 100$), 表示数据组数。

对于每组数据:

一行包含一个字符串, 格式为 $x^{\{y\}}$ 。其中 x, y 为正整数, 且满足 $1 \leq x, y \leq 10^{18}$ 。字符串中不包含任何其它字符。

输出格式

对于每组数据:

首先输出 10 行 10 个字符串, 表示字符画。再输出一行空白行, 以便分割各组数据。

样例

standard input
3 47^{2} 56^{2} 1^{1}
standard output
.....22222..... .4.....4.7777777.....2.....2222222.2222222.0000000.9999999. .4.....4.....7.22222.....2.....2.0.....0.9.....9. .4.....4.....7.2.....=====.....2.....2.0.....0.9.....9. .4444444.....7.22222.....2222222.2222222.0.....0.9999999.4.....7.....=====.....2.....2.....0.....0.....9.4.....7.....2.....2.....0.....0.....9.4.....7.....2222222.2222222.0000000.9999999.22222..... .5555555.6666666.....2.....3333333.....1.3333333.6666666. .5.....6.....22222.....3.....1.....3.6..... .5.....6.....2.....=====.....3.....1.....3.6..... .5555555.6666666.22222.....3333333.....1.3333333.6666666.5.6.....6.....=====.....3.....1.....3.6.....6.5.6.....6.....3.....1.....3.6.....6. .5555555.6666666.....3333333.....1.3333333.6666666.1.....1.....1.....1.1.....1.....1.1.....1.=====.....1.1.....1.....1.1.....=====.....1.1.....1.....1.....1.....

standard input
2 80~{1} 233~{233}
standard output
.....1..... .8888888.0000000.....1.....8888888.0000000. .8.....8.0.....0.....1.....8.....8.0.....0. .8.....8.0.....0.....1.=====.8.....8.0.....0. .8888888.0.....0.....1.....8888888.0.....0. .8.....8.0.....0.....=====.8.....8.0.....0. .8.....8.0.....0.....8.....8.0.....0. .8888888.0000000.....8888888.0000000.22222.33333.33333..... .2222222.3333333.3333333.....2.....3.....3.....IIIIIII.N.....N.FFFFFFFF.2.....3.....3.22222.33333.33333.....I....NN....N.F.....2.....3.....3.2.....3.....3.=====.I....N.N....N.F..... .2222222.3333333.3333333.22222.33333.33333.....I....N..N..N.FFFFFFFF. .2.....3.....3.....=====.I....N..N..N.F..... .2.....3.....3.....I....N....NN.F..... .2222222.3333333.3333333.....IIIIIII.N.....N.F.....

提示

普通数字字符:

.....0000000.....1.2222222.3333333.4.....4.5555555.6666666.7777777.8888888.9999999. .0.....0.....1.....2.....3.4.....4.5.....6.....7.8.....8.9.....9. .0.....0.....1.....2.....3.4.....4.5.....6.....7.8.....8.9.....9. .0.....0.....1.2222222.3333333.4444444.5555555.6666666.....7.8888888.9999999. .0.....0.....1.2.....3.....4.....5.6.....6.....7.8.....8.....9. .0.....0.....1.2.....3.....4.....5.6.....6.....7.8.....8.....9. .0000000.....1.2222222.3333333.....4.5555555.6666666.....7.8888888.9999999.

上标中的数字字符:

```
.....  
.00000.....1.22222.33333.4...4.55555.66666.77777.88888.99999.  
.0...0.....1.....2.....3.4...4.5.....6.....7.8...8.9...9.  
.0...0.....1.22222.33333.44444.55555.66666.....7.88888.99999.  
.0...0.....1.2.....3.....4.....5.6...6.....7.8...8.....9.  
.00000.....1.22222.33333.....4.55555.66666.....7.88888.99999.  
.....  
.....  
.....  
.....
```

符号:

```
.....  
.....  
.....IIIIIII.N....N.FFFFFFFF.  
.....I....NN....N.F.....  
.=====....I....N.N...N.F.....  
.....I....N..N..N.FFFFFFFF.  
.=====....I....N...N.N.F.....  
.....I....N....NN.F.....  
.....IIIIIII.N....N.F.....  
.....
```

Problem H. Travel Begins

输入文件: standard input

输出文件: standard output

时间限制: 1 second

空间限制: 256 megabytes

给定正整数 n, k , 我们称实数列 a_1, a_2, \dots, a_k 为关于 n 的 k -实数划分当且仅当 $\{a_i\}$ 满足下列条件:

- $\sum_{i=1}^k a_i = n$
- $\forall 1 \leq i \leq k : 0 \leq a_i \leq n$

记 n 的所有 k -实数划分构成的集合为 $D_k(n)$ 。

对于实数 x , 我们定义 $[x]$ 为不大于 x 的最大整数, 并定义 x 的小数部分为 $\{x\} = x - [x]$ 。定义四舍五入函数如下:

$$r(x) = \begin{cases} [x] & , \{x\} < 0.5 \\ [x] + 1 & , \{x\} \geq 0.5 \end{cases}$$

试求下述两式的值:

$$\min_{a \in D_k(n)} \left\{ \sum_{i=1}^k r(a_i) \right\}, \quad \max_{a \in D_k(n)} \left\{ \sum_{i=1}^k r(a_i) \right\}$$

输入格式

本题包含多组数据。

第一行包含一个整数 T ($1 \leq T \leq 10^5$), 表示数据组数。

对于每组数据:

一行包含两个正整数 n, k ($1 \leq n, k \leq 10^9$)。

输出格式

对于每组数据:

输出一行包含两个整数, 依次为 $\min_{a \in D_k(n)} \left\{ \sum_{i=1}^k r(a_i) \right\}$ 以及 $\max_{a \in D_k(n)} \left\{ \sum_{i=1}^k r(a_i) \right\}$ 的值。

样例

standard input	standard output
5	0 2
1 3	4 4
4 1	2 3
2 2	0 6
3 7	3 5
4 3	

Problem I. 数正方形

输入文件: standard input
输出文件: standard output
时间限制: 1 second
空间限制: 256 megabytes

“学霸题，数正方体，头顶标数法，从上往下数……”

你很快学会了怎么数正方体，所以现在想让你数正方形。

给出一个 $(2n+1) \times (2n+1)$ 的网格图，左下角的坐标为 $(0,0)$ ，右上角的坐标为 $(2n+1, 2n+1)$ 。在这个网格图中有 n 个矩形，第 i 个矩形左下角的坐标为 (x_{1i}, y_{1i}) ，右上角的坐标为 (x_{2i}, y_{2i}) ，且序列 $\{x_{11}, x_{12}, \dots, x_{1n}, x_{21}, x_{22}, \dots, x_{2n}\}$ 和序列 $\{y_{11}, y_{12}, \dots, y_{1n}, y_{21}, y_{22}, \dots, y_{2n}\}$ 分别都是 $1 \sim 2n$ 的一个排列。

我们按照以下方法对这 $(2n+1) \times (2n+1)$ 个格子进行黑白染色：被矩形覆盖了偶数次的格子染上白色，否则染上黑色。

求格子中有多少个 2×2 的正方形是纯色的，即仅由白色格子或黑色格子构成的正方形。

输入格式

第一行包含一个正整数 n ($1 \leq n \leq 10^5$)，表示矩形的个数。

接下来 n 行，每行包含四个正整数 $x_{1i}, y_{1i}, x_{2i}, y_{2i}$ ($1 \leq x_{1i} < x_{2i} \leq 2n, 1 \leq y_{1i} < y_{2i} \leq 2n$)，表示第 i 个矩形左下角和右上角的坐标。保证序列 $\{x_{11}, x_{12}, \dots, x_{1n}, x_{21}, x_{22}, \dots, x_{2n}\}$ 和序列 $\{y_{11}, y_{12}, \dots, y_{1n}, y_{21}, y_{22}, \dots, y_{2n}\}$ 分别都是 $1 \sim 2n$ 的一个排列。

输出格式

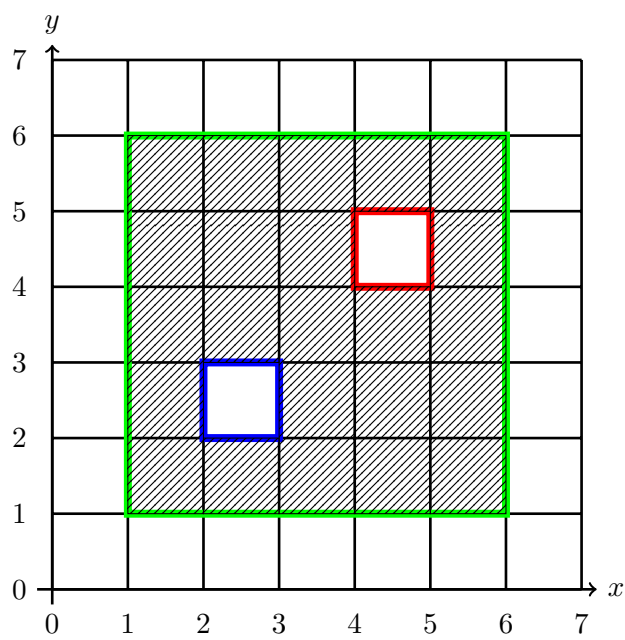
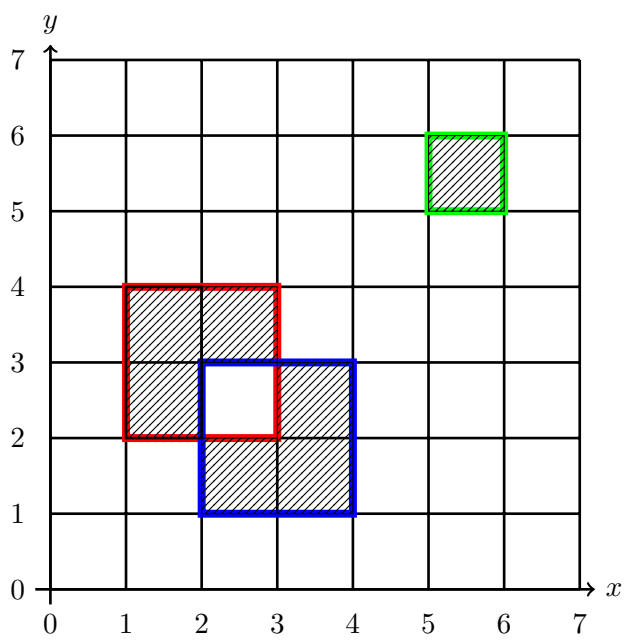
输出一行包含一个整数，表示答案。

样例

standard input	standard output
3 1 2 3 4 2 1 4 3 5 5 6 6	18
3 1 1 6 6 2 2 3 3 4 4 5 5	8

提示

两个样例染色后的网格图如下：

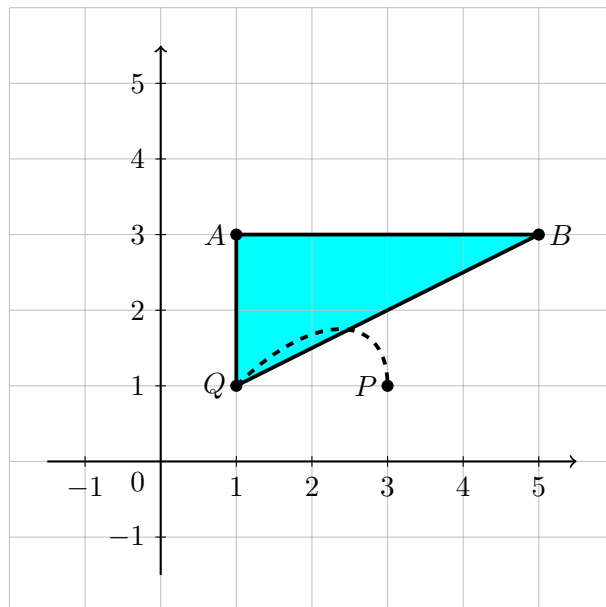


Problem J. Mocha 沉迷电子游戏

输入文件: standard input
 输出文件: standard output
 时间限制: 2 seconds
 空间限制: 256 megabytes

Mocha 最近沉迷电子游戏, 并且对 Talon 这一游戏角色产生了极大的兴趣。为了能够精通这一角色, Mocha 对该角色一项名为 Rake 的技能进行研究。

该技能允许 Talon 扔出一排刀刃, t 秒后刀刃会折返到 Talon 当前位置并对经过的敌人造成伤害。假设 Talon 当前位于点 P , 扔出的刀刃可以视作二维平面上的一条线段 AB 。Mocha 可以操纵 Talon 以每秒 v 的速度沿任意轨迹移动, 并且至多移动 t 秒。假设 t 秒后 Talon 所在位置为点 Q , 刀刃会折返到 Talon 当前位置 Q 并对 $\triangle QAB$ 内 (包括边界) 的敌人造成伤害。



如上图, Talon 起初位于点 $P(3, 1)$ 处, 刀刃可以视作端点分别为点 $A(1, 3)$ 和点 $B(5, 3)$ 的线段 AB 。之后 Mocha 操纵 Talon 沿图中虚线进行移动, 经过 $t_1 (t_1 \leq t)$ 秒后到达点 $Q(1, 1)$ 。又经过 $t - t_1$ 秒后, 刀刃返回到点 Q , 对 $\triangle QAB$ 的所有敌人造成伤害。

二维平面上的某个点为危险点, 当且仅当存在至少一种移动方案使得刀刃折返时能够对位于该点的敌人造成伤害, 即假设移动 t 秒后 Talon 所在位置为点 Q , 点 S 为危险点当且仅当点 S 位于 $\triangle QAB$ 的内部或者边界上。

但是 Mocha 的计算几何水平不高, 所以她限定了起点 P 和线段 AB 为以 AB 为底的非退化等腰三角形, 即 $PA = PB$ 且 $\triangle PAB$ 面积不为 0。Mocha 想知道对于给定的情况, 危险区域, 即所有危险点的并集的面积是多少, 但是 Mocha 已经开始了一局游戏, 于是她想请聪明的你帮她计算出答案。

输入格式

本题包含多组数据。

第一行包含一个整数 T ($1 \leq T \leq 2 \times 10^4$), 表示数据组数。

对于每组数据:

第一行包含两个整数 x_P, y_P ($-10^5 \leq x_P, y_P \leq 10^5$), 表示起点 P 的坐标。

第二行包含两个整数 x_A, y_A ($-10^5 \leq x_A, y_A \leq 10^5$), 表示线段端点 A 的坐标。

第三行包含两个整数 x_B, y_B ($-10^5 \leq x_B, y_B \leq 10^5$), 表示线段端点 B 的坐标。

第四行包含两个整数 v, t ($1 \leq v \leq 100, 1 \leq t \leq 10^5$), 表示 Talon 的移动速度和最长移动时间。

数据保证点 P, A, B 构成的三角形为以线段 AB 为底的非退化等腰三角形。

输出格式

对于每组数据:

输出一行包含一个浮点数, 代表危险区域的总面积。当且仅当你的答案和正确答案的绝对误差或相对误差不超过 10^{-6} 时, 你的答案会被视为正确, 即假设你的答案为 a , 正确答案为 b , 当且仅当 $\frac{|a-b|}{\max(1, |b|)} \leq 10^{-6}$ 时你的答案会被视为正确答案。

样例

standard input	standard output
5	54.785677060525
0 0	20.825663485551
-5 7	2242798580.838907851838
7 -5	21074117.679803092306
1 3	17853783044.606724914163
0 0	
-5 7	
7 -5	
1 1	
1 1	
5000 5000	
-4998 5000	
77 347	
13 41	
-2577 42	
-2577 40	
74 35	
0 0	
100000 0	
0 100000	
30 2357	

Problem K. 排列与质数

输入文件: standard input

输出文件: standard output

时间限制: 1 second

空间限制: 256 megabytes

给出一个正整数 n , 你需要构造一个 n 的排列 $P = (P_1, P_2, \dots, P_n)$, 满足:

- $\forall i \in [1, n], |P_i - P_{i \bmod n+1}|$ 为质数。

若有解, 请你输出任意一个满足条件的 n 的排列; 否则输出 -1 。

输入格式

第一行包含一个正整数 n ($2 \leq n \leq 10^5$)。

输出格式

一行包含若干整数, 表示答案。若包含一个整数 -1 , 表示题目无解; 若包含一个 n 的排列, 表示该排列满足题目要求。如果有多个排列满足要求, 你可以输出任意一种。

样例

standard input	standard output
5	4 1 3 5 2
2	-1

提示

对于样例一:

- $|P_1 - P_2| = 3$, 为质数。
- $|P_2 - P_3| = 2$, 为质数。
- $|P_3 - P_4| = 2$, 为质数。
- $|P_4 - P_5| = 3$, 为质数。
- $|P_5 - P_1| = 2$, 为质数。

因此这是其中一个满足条件的排列。

Problem L. 猜数游戏

输入文件: standard input

输出文件: standard output

时间限制: 6 seconds

空间限制: 256 megabytes

本题为交互题。

Oshwiciqwq 想跟你玩猜数游戏。这是一个关于分数的游戏，游戏里所有分数都是既约分数，即分子和分母互质的分数。

Oshwiciqwq 手里有一个隐藏的既约分数 $\frac{p}{q}$ ，满足 $1 \leq p \leq q \leq 10^9$ 。你只能向他询问恰好一次来确定这个分数。询问时你需要给出一个同样满足 $1 \leq a \leq b \leq 10^9$ 的既约分数 $\frac{a}{b}$ ，他会计算既约分数 $\left| \frac{p}{q} - \frac{a}{b} \right| = \frac{c}{d}$ 并告诉你 $c + d$ 的值。如果恰好有 $\frac{p}{q} = \frac{a}{b}$ ，那么他会告诉你 0。最后你要告诉他你猜出的 p, q 分别是多少。

Oshwiciqwq 会跟你玩 T 轮游戏。你必须猜错不超过 2 轮才能得到他的认可并通过此题。

输入格式

本题包含多组数据。

第一行包含一个整数 T ($1 \leq T \leq 10^5$)，表示数据组数（游戏轮数）。

对于每组数据（每轮游戏）：

本题没有初始输入，与交互器的交互方式参见交互格式。题目保证 $1 \leq p \leq q \leq 10^9$ 且 $\frac{p}{q}$ 是既约分数。题目保证第 i 轮游戏的 $\frac{p}{q}$ 是在你进行第 i 次询问之前选定的，不会根据第 i 次询问动态调整，但是可能根据你的前 $i - 1$ 次询问进行调整。

测试点的总数不超过 100。

交互格式

你只能询问恰一次以确定 $\frac{p}{q}$ 。

每一轮游戏开始时，你需要输出一行 `? a b` 表示你询问的既约分数。随后，交互器会向你返回一个非负整数，表示计算后的结果。

接下来，你需要输出一行 `! p q` 来提交你的答案。交互器会向你返回一个整数 0 或 1，其中 0 表示回答错误，1 表示回答正确。此时本轮游戏结束，若还有下一轮游戏，请继续处理。

你必须猜错不超过 2 轮才能通过此题。

你应当保证输出的 $\frac{a}{b}$ 和 $\frac{p}{q}$ 满足 $1 \leq a \leq b \leq 10^9, 1 \leq p \leq q \leq 10^9$ 且均是既约分数。若不满足，交互器的返回值将变为 -1。此时你必须终止程序，否则由于程序尝试从关闭的输入流获取输入等情况，你可能得到意料之外的评测结果。

注意，每次你输出的行末需要换行，并且在输出之后需要刷新输出流，否则你可能得到意料之外的评测结果。你可以通过以下方式刷新输出流：

- 对于 C++ 语言，使用 `fflush(stdout)` 或者 `cout.flush()`。

- 对于 C 语言，使用 `fflush(stdout)`。
- 对于 Java 语言，使用 `System.out.flush()`。
- 对于 Python 语言，需要载入 `sys` 模块，使用 `sys.stdout.flush()`。

样例

standard input	standard output
2	? 1 3
7	! 1 2
1	? 1 3
7	! 1 2
0	

提示

样例中进行了两轮游戏，答案分别是 $\frac{1}{2}$ 和 $\frac{1}{6}$ 。

两轮均询问了 $\frac{1}{3}$ ，且 $|\frac{1}{2} - \frac{1}{3}| = |\frac{1}{6} - \frac{1}{3}| = \frac{1}{6}$ ， $1 + 6 = 7$ ，所以交互器均返回计算结果 7。

两轮均猜测 $\frac{p}{q} = \frac{1}{2}$ 。第一轮交互器返回 1，代表猜测正确；第二轮交互器返回 0，代表猜测错误。