

A chmod

题目描述

chmod 是一个用于更改文件或目录权限的命令，它是 Linux 和其他类 Unix 操作系统中的常用命令之一。chmod 命令允许用户为文件或目录设置不同的权限，以控制谁可以读取，写入或执行这些文件。

在 Linux 系统中，每个文件或目录都有与之相关联的权限, 这些权限决定了谁可以对文件进行何种操作。用户被分为三类：所有者（owner），所属组（group）和其他人（others）。每类用户又都有读（r），写（w）和执行（x）三种权限。这 9 种权限可以分别指定。我们称**权限字符串**是一个长度为 9 的字符串，按顺序分别对应以上提到的 9 种权限，如果具有此权限，则为 r, w, x 中相应的那一个，否则为 -。

例如**权限字符串** rwxr-x--x 说明，该文件对于所有者具有全部权限，对于所属组用户只具有读和执行权限，而对于其他人只具有执行权限。

使用 chmod 命令时，你可以提供模式串来修改权限。在本题，我们只考虑长度为 3 的由不大于 7 的数字组成的模式串，其三个数字从左往右依次代表所有者（owner），所属组（group）和其他人（others）的权限。对于每个数字，其最低三个二进制位从高向低分别表示了该类用户是否具有读（r），写（w）和执行（x）权限。

例如：执行 chmod 760 file.txt 后, 表示文件的**权限字符串**为 rwxrw----。

给出若干合法的模式串，每次读入一条模式串后，你需要输出修改后文件的**权限字符串**。

输入格式

第一行一个正整数 T ($1 \leq T \leq 100$)，表示数据组数。

接下来 T 行，每行一条 chmod 模式串。

输出格式

共 T 行，表示模式串对应的**权限字符串**。

样例 #1

样例输入 #1

```
3
356
114
514
```

样例输出 #1

```
-wxr-xrw-
--x--xr--
r-x--xr--
```

B 表达式矩阵

题目描述

一个 $n \times m$ 的字符矩阵 a_{ij} ，被称为合法的表达式矩阵，当且仅当其满足如下条件：

- 矩阵只包含 '1', '+', '*' 字符。
- 对于矩阵的每行从左向右组成的字符串，均为合法的表达式。
- 对于矩阵的每列从上向下组成的字符串，均为合法的表达式。

一个合法的表达式矩阵的权值定义为，每行从左向右组成的字符串和每列从上向下组成的字符串共 $n + m$ 个表达式求值后的值求和的结果。

求所有 $n \times m$ 的合法表达式矩阵中，权值最小的那一个。如果有多个最小的答案，你可以给出任意一个。

我们定义字符串 s 是合法表达式如下：

- 如果 $s = \overbrace{111 \dots 111}^{\text{至少一个 } 1}$ ，则 s 是合法表达式。
- 如果 s 和 t 均为合法表达式，则 $s * t$ 也是合法表达式。
- 如果 s 和 t 均为合法表达式，则 $s + t$ 也是合法表达式。

输入格式

输入仅一行两个整数 n, m ($3 \leq n, m \leq 9$)，由空格隔开，为矩阵的行数和列数。

输出格式

输出共 n 行，每行 m 个字符，其中第 i 行的第 j 个字符为 a_{ij} ，为权值最小的矩阵。

如果有多个最小的答案，你可以给出任意一个。

样例 #1

样例输入 #1

```
4 4
```

样例输出 #1

```
1111
1*11
11*1
1111
```

提示

对于样例，此时矩阵的权值为 4488，可以证明不存在权值更小的矩阵。

C 换座位

题目描述

树王国在筹备着举办一次盛大的庆典！

Shirost 作为树王国的庆典设计师，准备邀请 n 个嘉宾来参加本次庆典。庆典上一共准备了 $2n$ 个座位，**一个座位最多只能坐一个人且一个人恰好坐一个座位**。Shirost 初步计划将第 i 个嘉宾安排在第 i 个座位上。但是总统调查了这 n 个嘉宾的意愿，第 i 个嘉宾的心仪座位为第 a_i 个座位。但除非能坐到心仪座位上，否则他们只愿意坐在原来的座位上。总统希望 Shirost 能够修改计划，使得尽可能多的嘉宾坐在他们的心仪座位上。

形式化的讲，你需要找到长为 n 的数组 b_i ($1 \leq i \leq n, 1 \leq b_i \leq 2n$) 满足 $\forall i \neq j, b_i \neq b_j$ 且 $\forall i, b_i = i$ 或 $b_i = a_i$ 。且最大化 $b_i = a_i$ 的个数。

你只需要输出最多的个数即可。

输入格式

输入第一行为一个整数 n ($1 \leq n \leq 10^5$)，表示总人数。

第二行 n 个整数 a_i ($1 \leq a_i \leq 2n$)，由空格隔开，表示每个人的心仪座位。

输出格式

输出仅一行一个整数，表示最多有多少嘉宾坐在他们的心仪座位上。

样例 #1

样例输入 #1

```
5
2 6 4 5 3
```

样例输出 #1

```
5
```

提示

所有人都可以换到自己的心仪座位。

D 双子序列

题目描述

小 L 看到不喜欢的字符串就很难受！看到它作为子序列出现也是！

给定一个长字符串 S 表示小 L 要阅读的文本，以及恰好两个短字符串 s_1, s_2 表示小 L 不想看到的字符串，三个字符串均由小写字母组成。

小 L 很反感这两个字符串作为子序列在文本内同时出现，他认为，一个字符串 T 的反感度为： s_1 作为 T 的子序列的出现次数，和 s_2 作为 T 的子序列的出现次数之积。

由于他要读 S 的每个子串，所以现在需要你求出 S 的所有子串的反感度值之和。由于答案可能过大，你只需要输出对 998244353 取模的结果。

定义一个字符串 H 是 T 的子串，当且仅当 H 由 T 删除最前面的若干字符和最后面的若干字符获得（前缀后缀可以一个字符都不删除，也可以把整个串全删除得到空串）。

定义一个字符串 H 是 T 的子序列，当且仅当 H 由 T 删除若干字符后获得（可以一个字符都不删除，也可以全删除后得到空子序列）。

输入格式

输入包括三行，每行一个仅由小写字母组成的字符串。

第一行的字符串代表 S ，第二行代表 s_1 ，第三行代表 s_2 。其中 $1 \leq |S| \leq 1 \times 10^5, 1 \leq |s_1|, |s_2| \leq 20$ 。

输出格式

输出一个整数代表求得的答案。

样例 #1

样例输入 #1

```
iccpicpc
icpc
ccpc
```

样例输出 #1

133

提示

子串起始位置	子串终止位置	icpc 次数	ccpc 次数
1	5	2	1
1	6	2	1
1	7	4	2
1	8	4	2
1	9	11	9
2	5	0	1
2	6	0	1
2	7	0	2
2	8	0	2
2	9	1	9
3	9	1	3
4	9	1	1
5	9	1	1
6	9	1	0

在其余的子串内，两个字符串作为子序列的出现次数均为 0。

答案为 $(2 \times 1) \times 2 + (4 \times 2) \times 2 + 11 \times 9 + (0 \times 1) \times 2 + (0 \times 2) \times 2 + 1 \times 9 + 1 \times 3 + (1 \times 1) \times 2 + 1 \times 0 = 133$ 。

E 商路

题目描述

一个圆周被 K 个点等分，这些点按照顺时针编号为 $1, 2, \dots, K$ 。其中点 a_1, a_2, \dots, a_n 分别建造有一座市场 M_1, M_2, \dots, M_n 。

一条从市场 i 出发，目标是市场 j 的商路是有向线段 M_iM_j ($i \neq j$) 且必须满足以下条件：

- 市场 j 必须是距离市场 i 最远的市场（如果有多个距离相同的最远的市场,那么任意一个均可）。
- 商路线段 M_iM_j 不能与其他商路线段在起点或者终点以外的地方相交或重合。

最多可以存在多少条商路？

输入格式

第一行包含两个整数 K, n ($3 \leq K \leq 10^9, 3 \leq n \leq \min(K, 10^5)$), 由空格隔开，表示有 n 个市场分布在圆周的 K 等分点上。

第二行包含 n 个整数 a_1, a_2, \dots, a_n ($1 \leq a_1 < a_2 < \dots < a_{n-1} < a_n \leq K$), 为建有市场的点的编号。

输出格式

输出一个整数表示最多可以存在的商路的数量。

样例 #1

样例输入 #1

```
10 5
1 2 5 6 8
```

样例输出 #1

```
2
```

样例 #2

样例输入 #2

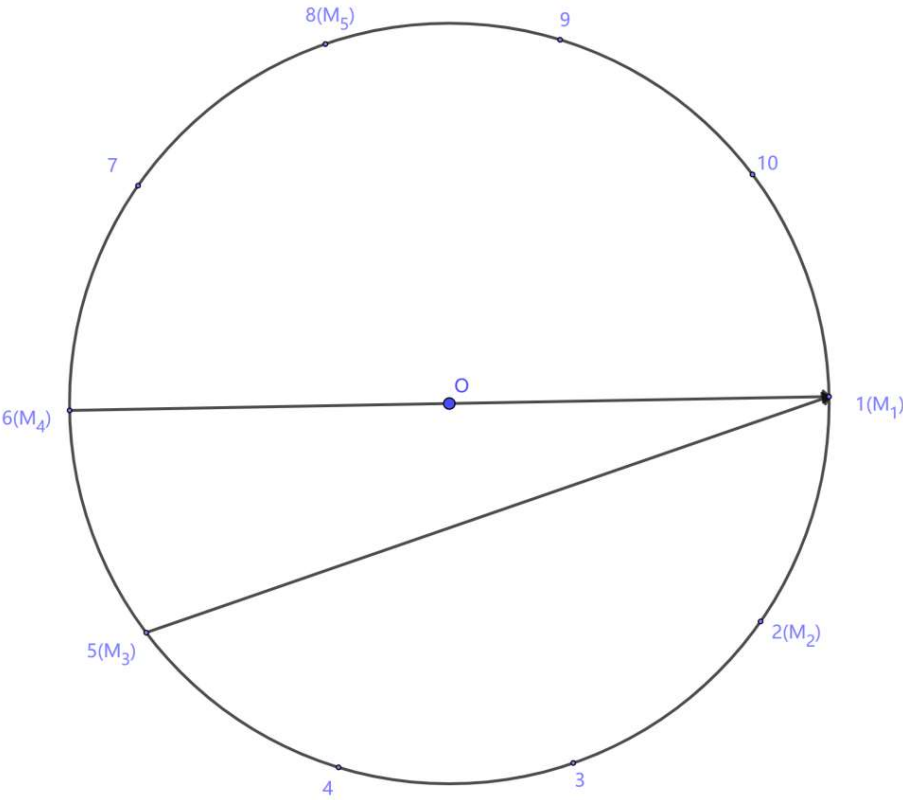
```
3 3
1 2 3
```

样例输出 #2

```
3
```

提示

对于第一个样例，其中两种可能的答案如下图所示：



样例 #1

样例输入 #1

```
2
3
10 4 15
4
10 10 5 4
```

样例输出 #1

```
15
10
```

提示

对于第一组数据，只需将第三份代码提交两次即可。

对于第二组数据，提交前两份代码即可。

G 消失的数字

题目描述

uuku 正在学习数位 DP！

但某一天，他发现 $1, 2, \dots, 9$ 这九个数字中的 x 消失了，也就是说所有包含 x 这个数字的数都消失了。

这让他非常惊慌，因为这对统计数的个数有很大的影响。

现在，他希望你能帮助他求出这种情况下数 n 在未消失的自然数中从小到大排在第几位。

输入格式

本题有多组测试数据，输入第一行为一个整数 T ($1 \leq T \leq 10^5$)，表示测试数据的组数。

对于每组测试数据：

共一行两个整数 n, x ($0 \leq n \leq 10^{18}, 1 \leq x \leq 9$)，由空格隔开，含义如题目描述所述。

数据保证 n 中不包含数字 x 。

输出格式

对于每组测试数据：输出一行一个整数，表示数 n 的排名。

样例 #1

样例输入 #1

```
5
9 4
99 7
12345678 9
9475632111234123 8
998244353114514 7
```

样例输出 #1

```
9
81
6053445
1758041005111510
205404686678741
```

提示

对于样例的第一组数据，自然数列变为 0, 1, 2, 3, 5, 6, 7, 8, 9, 其中数 9 是第 9 个。

H 最大流

题目描述

给定一个 n 个点 m 条边的有向无环图，图中每条边的容量为 1。对点 1 以外的每个点 i ，设从点 1 到点 i 的最大流为 f_i ，试求出 $\min\{f_i, k\}$ 。

在边容量为 1 的图上，一个从点 1 到点 i 的流即为一条从点 1 到点 i 的路径。如果从点 1 到点 i 最多能同时有 f_i 个不交的流（即没有一条边同时属于两个流），则认为点 1 到点 i 的最大流是 f_i 。

输入格式

输入第一行为三个整数 n, m, k ($2 \leq n \leq 10^5, 1 \leq m \leq 2 \times 10^5, 1 \leq k \leq 50$)，由空格隔开，为图的点数，边数和参数。

接下来 m 行，每行两个整数 x_i, y_i ($1 \leq x_i, y_i \leq n, x_i \neq y_i$)，由空格隔开，描述一条有向边。

图中保证没有自环，但是可能存在重边，保证给出的是一个有向无环图。

输出格式

输出仅一行 $n - 1$ 个整数，由空格隔开。对于第 i 个整数，如果从结点 1 到结点 $i + 1$ 的最大流不超过 k ，则为最大流的值，否则为 k 。

样例 #1

样例输入 #1

```
7 12 3
1 2
1 3
3 2
3 4
2 4
1 5
5 6
3 6
1 7
5 7
6 7
4 7
```

样例输出 #1

```
2 1 2 1 2 3
```


样例 #2

样例输入 #2

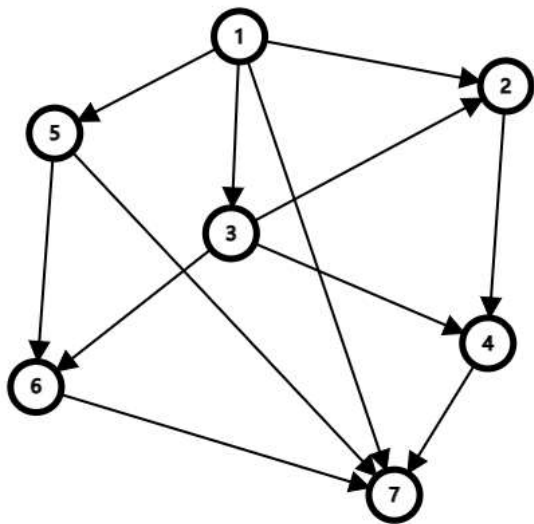
```
5 8 50
1 2
1 2
1 2
3 2
2 4
2 4
2 4
2 4
```

样例输出 #2

```
3 0 3 0
```

提示

第一个样例所述图如下：



我们可以找到 4 条从点 1 到点 7 的不相交路径：

- 1->7
- 1->5->7
- 1->3->6->7
- 1->2->4->7

我们无法找到更多条从点 1 到点 7 的不相交路径：

所以点 1 到点 7 的最大流为 $f_7 = 4$ ，但是因为 $k = 3$ ，所以答案的第六个整数为 3。

I 猜质数 I

题目描述

这是一道交互题。

MCPlayer542 手上有一个神秘的**奇质数** p ，但他并不想让你知道这个数是多少。

他打算用一个函数 $f(x)$ 来加密他的数，其值为 x 在十进制下的各位数字之和，例如 $f(5) = 5$, $f(542) = 5 + 4 + 2 = 11$, $f(1024) = 1 + 0 + 2 + 4 = 7$ 。

然而考虑到你太聪明，他想了想，决定把加密函数改成：

$$g(x) = f(f(f(f(f(f(f(f(f(x))))))))))$$

即连续应用 10 次 $f(x)$ ，并把手上的 p 换成了 $q = p^k$ 。

现在他准备给你 n 个整数 $g(q^{a_1})$, $g(q^{a_2})$, \dots , $g(q^{a_n})$ ，并希望你能告诉他

$$q^{a_1} \bmod (m \cdot a_1), q^{a_2} \bmod (m \cdot a_2), \dots, q^{a_n} \bmod (m \cdot a_n)$$

分别是多少。他觉得你肯定猜不到，所以决定让你自己选择 m 和 a_1, a_2, \dots, a_n 。你能完成这个任务吗？

注意： m 的范围有特殊限制。

输入格式

输入包含多组测试数据。数据的第一行包含一个整数 t ($1 \leq t \leq 500$)，表示数据组数。每组数据的交互流程在下文中描述。

在每组数据中，输入的第一行包含两个整数 n 和 k ($1 \leq n \leq 50$, $1 \leq k \leq 10^9$)，用单个空格分隔，含义见题目描述。

接下来每次交互，输出一行一个整数 a_i ($1 \leq a_i \leq 10^{18}$, a_i 互不相同)，表示要询问的 q 的幂次。

给出一个询问后，你应该读入一个整数，即 $g(q^{a_i})$ 。

你需要在进行完 n 轮交互之后输出一行一个整数 m ($m \geq 35$, $1 \leq m \cdot a_i \leq 10^{18}$)，再输出一行 n 个数，即 $q^{a_1} \bmod (m \cdot a_1)$, $q^{a_2} \bmod (m \cdot a_2)$, \dots , $q^{a_n} \bmod (m \cdot a_n)$ ，用单个空格分隔，表示答案。

你必须进行恰好 n 轮交互，随后输出答案并结束程序，否则你可能得到无法预测的结果。

注意在你的程序每轮输出结束时（即，每一次交互输出 a_i 时和最后输出 m 与答案时）需要输出回车并刷新输出缓冲区，否则你将会得到 Idleness Limit Exceeded。

- C 的 `fflush(stdout)`;
- C++ 的 `cout.flush()`;
- Java 的 `System.out.flush()`;
- Python 的 `stdout.flush()`;

来刷新输出缓冲区。

保证在每组数据中的奇质数 p ($2 < p \leq 10^{18}$) 都是在交互前确定的，即不会随着你的输入而变化。

如果你最后输出的答案正确，你会得到 Accepted;

如果你输出的 m 或 a_i 不符合题目范围要求，或最后输出的答案不正确，你会得到 Wrong Answer。

此外，其他的评测结果仍会在评测过程中根据通常情况返回。

输出格式

样例 #1

样例输入 #1

```
2
3 1

3

9

9

3 2

4

4

4
```

样例输出 #1

```
1

2

3

100
3 9 27

1

7

49

49
0 0 0
```

提示

在第一组数据中，MCPlayer542 手上的奇质数 $p = 3$ ，因此有 $q = p^k = 3$ 。

我们选择数组 $a = \{1, 2, 3\}$ ，依次得到 $g(3^1) = 3, g(3^2) = 9, g(3^3) = 9$ 。

随后我们猜到 $p = 3$ ，选择 $m = 100$ ，因此输出 $3^1 \bmod (100 \times 1) = 3, 3^2 \bmod (100 \times 2) = 9, 3^3 \bmod (100 \times 3) = 27$ 。

在第二组数据中，MCPlayer542 手上的奇质数 $p = 7$ ，因此有 $q = p^k = 49$ 。

我们选择数组 $a = \{1, 7, 49\}$ ，依次得到 $g(49^1) = 4, g(49^7) = 4, g(49^{49}) = 4$ 。

随后我们**敏锐地发现** $p = 7$ ，选择 $m = 49$ ，因此输出 $49^1 \bmod (49 \times 1) = 0, 49^7 \bmod (49 \times 7) = 0, 49^{49} \bmod (49 \times 49) = 0$ 。

注：第二组数据中的“**敏锐地发现**”仅作为交互流程的示意，并不保证上述交互可以确定 $p = 7$ 。

J 猜质数 II

题目描述

为了悄悄准备一个神秘的质数，MCPlayer542 伤透了脑筋。

随后他发明了一种聪明愚蠢的办法，并起名为“质数分”。

他准备了 n 个不同的数 a_1, a_2, \dots, a_n 作为测试点，并定义“质数分” $score(x, l, r)$ 如下：

$$score(x, l, r) = \sum_{i=l}^r f(x, a_i)$$

其中

$$f(x, y) = \begin{cases} u - y, & x = 1 \\ u, & 1 < x \leq y, \gcd(x, y) = 1 \\ -x \cdot y, & x \neq 1, \gcd(x, y) = x \\ 0, & \text{otherwise} \end{cases}$$

可见质数的“质数分”通常会比较高但还是没什么卵用。

于是 MCPlayer542 急了，现在他只想暴力乱测，并得到一个得分和 $\sum_{i=1}^{10^6} score(i, l, r)$ 。他打算测试 q 次，每次测试给定 u 和 l ，其中 u 为函数 $f(x, y)$ 的参数。

对于每次询问，他想知道所能得到的最大得分和以及能得到这个最大得分和的最小 r 是多少。

输入格式

输入数据的第一行包含两个整数 n, q ($1 \leq n, q \leq 5 \times 10^5$)，用单个空格分隔。

第二行包含 n 个整数 a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^6$)，用单个空格分隔，表示测试点的数据。

接下来 q 行，每行两个整数 u_i, l_i ($1 \leq u_i \leq 1.8 \times 10^7, 1 \leq l_i \leq n$)，用单个空格分隔，表示一次询问。

输出格式

对每个询问输出一行两个数，即对应询问的最大得分和以及能获得该得分的最小 r_i ($l_i \leq r_i \leq n$)，用单个空格分隔。

样例 #1

样例输入 #1

```
10 7
10 9 2 1 5 3 10 10 1 8
14 4
17 5
13 10
16 1
12 4
16 6
16 3
```

样例输出 #1

```
55 6
60 6
-68 10
-58 6
41 6
20 6
79 6
```

样例 #2

样例输入 #2

```
6 8
3 7 7 10 8 9
21 1
20 4
21 3
21 5
21 1
21 2
21 2
21 5
```

样例输出 #2

```
170 3
-100 4
70 3
-27 6
170 3
140 3
140 3
-27 6
```

提示

在样例 1 的第一个询问中， $u_1 = 14$ ， $l_1 = 4$ 。若我们选择 $r_1 = 6$ ，则最后的得分和为 $\sum_{i=1}^{10^6} score(i, 4, 6)$ 。其中：

- $score(1, 4, 6) = 13 + 9 + 11 = 33$;
- $score(2, 4, 6) = 0 + 14 + 14 = 28$;
- $score(3, 4, 6) = 0 + 14 - 9 = 5$;
- $score(4, 4, 6) = 0 + 14 + 0 = 14$;
- $score(5, 4, 6) = 0 - 25 + 0 = -25$;
- i 取其他值时均有 $score(i, 4, 6) = 0 + 0 + 0 = 0$ 。

故 $r_1 = 6$ 时得分和为 $33 + 28 + 5 + 14 - 25 = 55$ 。

可以证明在 r_1 取其他值时无法得到更大的得分和，故答案为 55，且能达成的最小 r_1 为 6。

K 致命公司

题目描述

Shirost 最近沉迷于一款名为《致命公司》的游戏。

在游戏中，玩家将作为“公司”的合同工，从废弃的工业星球收集废料。在探索的过程中，Shirost 将遇到一个名为弹簧头的怪物，弹簧头会在无人注视它时快速移动，但在被注视时则会保持不动。一旦接近，弹簧头就会立刻杀死 Shirost。

现在 Shirost 站在 n 个无限长的通道的交汇点，他知道有 m 个弹簧头，分别会在 t_i 时刻出现在 x_i 号通道距离他 y_i 米远处。

对于每个时刻 j ，以下三个事件将会依次发生：

- 1. 在该时刻开始时，满足 $t_i = j$ 的弹簧头会出现在 x_i 号通道距离 Shirost y_i 米远处。
- 2. Shirost 选择凝视任意一个通道。
- 3. 该通道内的所有弹簧头无法移动，而其他通道中**已经出现**的弹簧头会向他移动 k 米，如果某个弹簧头到达他所在位置，则他将在时刻 j 死亡。

Shirost 想知道他最晚可以活到哪个时刻。换句话说，如果 Shirost 最晚在时刻 j 死亡，那么你需要输出 $j - 1$ 。如果他不会死亡，则输出 -1 。

输入格式

输入第一行为三个整数 n, m, k ($1 \leq n \leq 5 \times 10^5, 1 \leq m \leq 5 \times 10^5, 1 \leq k \leq 10^{18}$)，由空格隔开，为无限长通道的个数，弹簧头的个数，弹簧头每时刻移动的距离。

接下来 m 行，每行三个整数 t_i, x_i, y_i ($1 \leq t_i \leq 10^{18}, 1 \leq x_i \leq n, 1 \leq y_i \leq 10^{18}$)，由空格隔开，描述弹簧头出现的时刻，出现的通道编号，距离 Shirost 有多远。

输出格式

输出仅一行一个整数，表示 Shirost 最晚可以活到哪个时刻。如果他不会死亡，则输出 -1 。

样例 #1

样例输入 #1

```
2 3 2
1 1 6
2 2 7
3 1 8
```

样例输出 #1

```
6
```

样例 #2

样例输入 #2

```
114514 6 1919810
1 1 1
1 1 9
1 4 1
1 5 9
1 1 8
1 4 10
```

样例输出 #2

```
0
```

提示

Shirost 可以按如下所述行动：

时刻	凝视的通道	弹簧头 1 距离	弹簧头 2 距离	弹簧头 3 距离
1	1	6		

时刻	凝视的通道	弹簧头 1 距离	弹簧头 2 距离	弹簧头 3 距离
2	1	6	5	
3	2	4	5	6
4	1	4	3	6
5	2	2	3	4
6	1	2	1	4

在第 7 时刻，无论看哪个通道，Shirost 都会在该时刻内被弹簧头杀死。所以答案是 6。

L 下棋

题目描述

LNC 喜欢所有 k 进制下所有数位的乘积为自身因子的数。他称之为 LNC 数。例如：

当 $k = 10$ 时, $y = (36)_{10}$ 是 LNC 数, 因为 $(3 \times 6) \mid 36$ 。

当 $k = 4$ 时, $y = (12)_4$ 是 LNC 数, 因为转换成十进制后 $(12)_4 = (6)_{10}$, 而 $(1 \times 2) \mid 6$ 。

当 $k = 2$ 时, $y = (1101)_2$ 不是 LNC 数, 因为转换成十进制后 $(1101)_2 = (13)_{10}$, 而 0 不是 13 的因子。

LNC 在和 LJJ 玩一个游戏, LJJ 给出 x 枚棋子, 然后 LNC 选定一个整数 $k (k \geq 2)$ 。随后他们交替取走若干枚棋子, 要求取走的棋子数量是 k 进制意义下的 LNC 数。LNC 先手, 先取完的获胜。两个人都绝顶聪明, 故都会选择最优的策略。

LJJ 觉得这个游戏很不公平, 他们一共玩了 T 局游戏, 对于每局游戏他给出的 x , 他希望知道最小的 k 使得 LNC 先手必胜。

输入格式

输入第一行一个正整数 $T (1 \leq T \leq 1 \times 10^2)$, 表示数据组数。

接下来 T 行每行一个正整数 $x (3 \leq x \leq 10^{18})$, 表示 LJJ 给出的数 x 。

输出格式

输出 T 行每行一个正整数 k , 表示每个询问的最小的 k , 使 LNC 先手必胜。

样例 #1

样例输入 #1

```
3
9
5
10
```

样例输出 #1

```
2
2
3
```

提示

当 $x = 5$ 的时候, LNC 可以选择 $k = 2$ 。 $x = (5)_{10} = (101)_2$ 。

LNC 先手拿掉 $(11)_2$, 此时 $x = (10)_2$, LJJ 只能拿走 $(1)_2$, LNC 拿走最后的 $(1)_2$ 获胜。

又因为 $k = 2$ 已经不能再小了，所以最终答案为 $k = 2$ 。

M 窗花

题目描述

有一扇 $100\text{cm} \times 100\text{cm}$ 的窗户和 n 个对角线长为 2cm 的正方形窗花。建立坐标系，以窗户左下角的坐标为原点 $(0, 0)$ ，右上角坐标为 $(100, 100)$ ，第 i 个窗花中心被贴在非边缘的整坐标点 (x_i, y_i) ($1 \leq x_i, y_i \leq 99$) 上，窗花的对角线与坐标轴平行。

问窗户有多大的面积被至少一片窗花覆盖。

输入格式

第一行一个整数 n ($1 \leq n \leq 10000$)。

接下来 n 行，每行两个整数 x_i, y_i ($1 \leq x_i, y_i \leq 99$)，含义如上所述。

输出格式

输出仅一行一个实数，为被至少一片窗花覆盖的面积。

你的答案被认为正确当且仅当其相对误差或绝对误差不超过 10^{-4} 。形式化地说，假设你的答案是 a ，标准答案是 b 。你的答案被认为正确当且仅当 $\frac{|a-b|}{\max(1, |b|)} \leq 10^{-4}$ 。

样例 #1

样例输入 #1

```
5
1 1
2 1
3 2
5 5
5 5
```

样例输出 #1

```
7.5
```

提示

对于第一个样例的解释如图：

