



问题 A. 足球比赛

输入文件: 标准输入 输出文件: 标准
输出时间限制: 2 秒 内存限制: 512
MB

作为一名体育迷, 你作为鸽子王国的主要领导者, 正在组织一场足球比赛! 共有 N 名玩家报名参加比赛, 您计划将他们分为三组: 红队、蓝队和观众。红队和蓝队的球员人数可以不同。

N 参与者中有 M 对朋友, 其中对于给定的常数 $K \geq 1$, 其中 $M \geq 2KN$ 。友谊是相互的, 这意味着如果 a 是 b 的朋友, 则 b 是 a 的朋友, 反之亦然。为了让比赛更加精彩, 您需要确保红队中的每个玩家在蓝队中至少有 $K + 1$ 个好友, 并且蓝队中的每个玩家在红队中至少有 $K + 1$ 个好友。你能找到满足这些限制的安排吗?

输入

第一行包含一个整数 T ($1 \leq T \leq 50\,000$), 表示测试用例的数量。对于每个测试用例:

第一行包含三个整数 N 、 M 和 K ($1 \leq N, M, K \leq 50\,000$ 和 $M \geq 2KN$), 分别表示玩家数量、好友对数量和给定常数。

然后是 M 行, 每行包含两个整数 u 和 v ($1 \leq u < v \leq N$), 表示 u 和 v 是友元。

保证在每个测试用例中, 每对 (u, v) 最多出现一次, 并且所有测试用例的 M 之和不超过 $50\,000$ 。

输出

对于每个测试用例, 输出两行:

第一行以一个整数 R 开始, 表示红队的球员人数。然后是 R 个空格分隔的整数, 每个整数表示红队中一名球员的索引。

第二行遵循相同的格式。它以整数 B 开头, 表示蓝队的球员人数。然后是 B 个空格分隔的整数, 每个整数表示蓝队中一名球员的索引。

如果有多个解, 可以输出其中任意一个。可以证明, 在这样的情况下
h 有有限制, 解决方案总是存在的。



例子

<i>standard input</i>	<i>standard output</i>
2	3 2 3 4
5 10 1	2 1 5
1 2	3 2 8 10
1 3	2 1 9
1 4	
1 5	
2 3	
2 4	
2 5	
3 4	
3 5	
4 5	
10 20 1	
1 2	
2 3	
3 4	
4 5	
5 6	
6 7	
7 8	
8 9	
9 10	
1 10	
1 4	
4 7	
7 10	
3 10	
3 6	
6 9	
2 9	
2 5	
5 8	
1 8	



问题B.扭蛋

输入文件: 标准输入 输出文件: 标准
输出时间限制: 5 秒 内存限制: 512
MB

根据维基百科，“扭蛋游戏是一种实现扭蛋（玩具自动售货机）机制的视频游戏”。与战利品盒类似，扭蛋游戏诱导玩家花费游戏内货币来接收随机虚拟物品。

其中一种扭蛋游戏称为“Step-up Gacha”，这意味着玩家每次滚动时滚动稀有物品的机会都会增加。例如，现象级游戏《原神》确保您始终可以在任意连续十次滚动中抽出四星级物品或角色。

如果我们对这些滚动规则进行抽象，将会很有帮助。考虑一款包含 0 星、1 星、...、 m 星项目的游戏。假设单次抽出 i 星物品的概率为 $\sum_{j=0}^i a_j$ 。一个单次抽奖是级别 0 滚动，级别 k 的滚动恰好由级别 $(k-1)$ 滚动的 b_k 轮组成。滚动的最高级别是 n 。

如果级别 k 滚动确保满足以下条件，则该级别是合法的：

- 至少绘制了一项至少具有 k 颗星的项目，
- 对于它包含的所有 b_k 级别 $(k-1)$ 滚动，至少绘制一个具有至少 $(k-1)$ 星的项目，
- ...依此类推，一直到每个 0 级滚动（这是一次抽奖），至少会抽出一个至少有 0 颗星的物品。

令 p_i 为从合法的 n 级滚动中抽取的 i 星项目的预期数量，并令 q 为 n 级滚动合法的概率。找到值 p_i 和 q 。为了避免令人不快的巨大数字和被零除，对于所有 $0 \leq i \leq m$ ，您应该只输出值 $(p_i \cdot q) \bmod 998\,244\,353$ 。

输入

第一行包含两个整数 m 和 n : 星星的最大数量和滚动的最高级别 ($1 \leq n \leq m \leq 4000$)。

第二行包含 $m + 1$ 个整数 a_0, a_1, \dots, a_m : 滚动具有 $0, 1, \dots, m$ 星 ($1 \leq a_i \leq 4000$) 的项目的频率。

第三行包含 n 整数 b_1, b_2, \dots, b_n : 1 级, 2, ..., n ($2 \leq b_i \leq 4000$) 滚动中先前级别滚动的数量。

输出

输出 $m + 1$ 行。第 i 行应包含一个整数: $(p_{i-1} \cdot q) \bmod 998\,244\,35$ 的值

3。



例子

<i>standard input</i>	<i>standard output</i>
2 1 1 1 1 3	554580197 1 1
2 1 89 10 1 10	989586456 1 299473306
3 2 1 1 2 1 2 3	58137752 260406016 517809313 758026833

笔记

在第一个示例中，有理形式的答案是： $\frac{8}{9}, 1, 1$ 。

-



问题 C. 调查

输入文件: 标准输入 输出文件: 标准
输出时间限制: 1 秒 内存限制: 512
MB

您正在进行一项调查，并希望在您的社交群体中找到受访者。您的社交群组规模为 n ，您的预算为 m 美元。您需要将 m 美元分成 n 股。每个团体成员将统一随机获得一份份额。请注意，每股包含的资金可以是任何非负实数。

幸运的是，您知道每个小组成员的奖励阈值。如果一个人的奖励阈值为 x ，当且仅当他们收到至少 x 美元的份额时，他们才会参与调查，否则他们将只接受付款而不参与调查。由于您希望尽可能多的小组成员参与调查，因此您需要设计一个计划，将 m 美元分成 n 份，以便最大限度地增加参与调查的预期成员数量。

输入

第一行包含两个整数 n ($1 \leq n \leq 1000$) 和 m ($1 \leq m \leq 5000$)，表示群组成员数量和您的预算。

下一行包含 n 整数 x_1, x_2, \dots, x_n ($0 \leq x_i \leq m$)，表示每个成员的奖励阈值。

输出

打印单个实数：预计参与调查的最大会员人数。

如果输出与陪审团答案之间的绝对或相对误差最多为 10^{-9} ，则答案将被视为正确。

例子

<i>standard input</i>	<i>standard output</i>
5 5 1 2 3 3 4	1.200000000000
8 69 10 10 10 10 10 4 5	6.375000000000



问题D.站

输入文件: 标准输入 输出文件: 标准
输出时间限制: 4.5 秒 内存限制: 102
4 MB

A市主要街道沿线有 n 个公交车站和 n 条公交线路。公交车站从左到右标记为1到 n , i 站的重要性为 a_i 。公交线路的编号也从 1 到 n 。 k 线的公交车停靠重要性大于或等于 k 的车站。每条公交线路都有双向运营。

站在 x 站的游客可以乘坐任何停靠在 x 站的巴士, 选择一个方向, 然后前往该巴士在该方向所经过的下一个站 y (当然, 只有存在这样的站才有可能)。如果 $y < x$, 则该行程的费用为 l_x 元; 如果 $y > x$, 则为 r_x 元。游客可以乘坐多次巴士到达目的地。

现在有 q 个游客, 第 j 个游客想要从 s_j 站前往 t_j 站。你的任务是找出每个游客的最小路线成本。

保证对于从 1 到 $n - 1$ 的每个 i , 以下条件成立: $l_i \leq l_{i+1}$ 和 $r_i \geq r_{i+1}$ 。

输入

输入的第一行包含一个整数 T , 即测试用例的数量 ($1 \leq T \leq 3 \cdot 10^4$)。测试用例的描述如下。

每个测试用例的第一行包含两个整数 n 和 q : 车站数量和游客数量 ($1 \leq n, q \leq 3 \cdot 10^5$)。

第二行包含 n 整数 a_1, \dots, a_n , 其中 a_i 是站点 i ($1 \leq a_i \leq n$) 的重要性。

然后跟随 n 行, 其中第 i 行包含两个整数 l_i 和 r_i : 站点 i ($1 \leq l_i, r_i \leq 10$) 的成本 $l_i \leq l_{i+1}$ 、 $r_i \geq r_{i+1}$ 。

然后沿着 q 行走, 其中第 j 行包含两个整数 s_j 和 t_j : 第 j 条游客 ($1 \leq s_j, t_j \leq n$) 的路线端点。

所有测试用例的 n 之和与 q 之和不超过 $3 \cdot 10^5$ 。

输出

对于每个游客, 输出一行包含答案。



例子

<i>standard input</i>	<i>standard output</i>
1	33
9 6	9
1 7 3 4 9 9 1 2 2	6
1 11	8
1 11	17
5 11	0
7 10	
8 6	
8 4	
8 3	
9 1	
10 1	
1 9	
5 1	
3 1	
7 6	
2 6	
1 1	



问题 E. 猜数

输入文件: 标准输入 输出文件: 标准
输出时间限制: 1 秒 内存限制: 512
MB

Alice 有一个秘密整数 y , 它是从 $[1, 10^{18}]$ 中选择的。鲍勃想要得到这个号码, 所以他问了爱丽丝一些问题。在每个问题中, Bob 将 $[1, 10^{18}]$ 中的整数 x 提供给 Alice, Alice 如果 $y < x$ 返回 0, 如果 $y = x$ 返回 1, 否则返回 2。

一切听起来都很棒, 但是爱丽丝和鲍勃之间的通讯却被伊芙篡改了! Eve 编写了一个随机数生成器 \mathcal{H} , 它生成 0 到 $n - 1$ 之间的伪随机数。每次 Bob 给出该数字时, Eve 都会调用 \mathcal{H} 生成伪随机数 x , 并将 x 与 Alice 返回的结果按位异或和提供给 Bob。

Bob 发现 Alice 返回的结果已被某人篡改, 因为他得到的结果相互矛盾。Bob 通过某种方式获得了 \mathcal{H} 的源代码, 如下所示。

Algorithm 1: random number generator \mathcal{H}

Data: constant n , constant P , seed of generator *seed*
Result: a number x selected from $[0, n - 1]$
1 $seed \leftarrow (seed \times n) \bmod P$;
2 $result \leftarrow seed \bmod n$;
3 **return** *result*;

图 1: \mathcal{H} 的工作原理。

常量 P 和 n 在代码中指定, 因此 Bob 也知道它们的值。但他对 *seed* 一无所知, 只是知道篡改者 Eve 在交互过程中不会修改该值。你能帮助鲍勃在 100 次查询中得到爱丽丝秘密号码的值吗?

输入

第一行包含一个整数 t , 即测试用例的数量 ($1 \leq t \leq 100$)。测试用例的描述如下。

每个测试用例都以包含两个整数 n 和 P 的行开头: \mathcal{H} 中的两个常量。保证 $3 \leq n \leq 4$ 、 $10 \leq P \leq 10^{18}$ 和 P 是素数。

交互协议

要提出问题, 请打印一行, 格式为 “? x ” ($1 \leq x \leq 10^{18}$)。然后, 你应该阅读一行的答案。在每个测试用例中, 您最多可以提出 100 个问题。

如果答案是非负整数, 那就是 Bob 收到的数字, 你可以继续猜测。如果答案是 -1, 则表示您超出了查询次数或进行了无效查询。收到 -1 后立即退出, 您将看到“错误答案”的判决。否则, 您可以获得任意判决, 因为您的解决方案将继续从关闭的流中读取。

如果您获得了秘密号码 y , 请打印一行, 格式为 “! y ”。然后, 你应该阅读一行的答案。

如果答案是 -1, 则说明你的猜测是错误的, 你应该立即退出。如果答案是 1, 则说明你的猜测是正确的, 你应该阅读并解决下一个测试用例。请注意, 打印此秘密数字不算作 100 个查询之一。

确保交互器是确定性的, 这意味着交互器不会根据您的查询动态修改 y 和 *seed* 的值。

不要忘记打印行尾字符并在每次解决方案打印一行时刷新输出。否则, 您可能会得到“超出空闲限制”的判决。



例子

<i>standard input</i>	<i>standard output</i>
1	
3 998244353	0
? 4	0
? 2	1
? 1	1
! 1	

笔记

在示例中，我们有 $seed = 0$ ，因此 Bob 总是从 Alice 那里收到正确的答案。Bob 在这个测试用例中直接使用二分查找是合理的，但它不适用于其他可能的情况。



问题 F. 建造一座城市

输入文件: 标准输入 输出文件: 标准
输出时间限制: 3 秒 内存限制: 1024
MB

最近, 你沉迷于一款名为“Build a City”的游戏。

该游戏可以在二维地图上进行。地图上有 $n + 1$ 个人类住区, 标记为0到 n , 其坐标为 (x_i, y_i) , 即 $i = 0, 1, \dots, n$ 。保证 $x_0 = y_0 = 0$, 即定居点0位于原点。

你的城市是一个由墙组成的矩形, 其边与坐标轴平行。当该定居点位于矩形内部或矩形边界时, 则称该城市包含一个定居点。

首先, 你的城市包含定居点0, 并且在原点处是一个退化矩形。您只需一步即可获得一处无人居住的定居点, 并建造一些围墙将其围在城市中。具体来说, 新城市是边与坐标轴平行的最小矩形, 因此它包含旧城市所包含的内容以及新获得的定居点。你的目标是包围你城市中的所有人类住区。

您所在城市的基础设施部门一次性只能建造总长度不超过 m 的墙。请注意, 现有的墙壁可以重复使用, 但不能放置在其他地方。因此, 每次移动所建造的墙的长度是新矩形的周长减去原始矩形和新矩形的公共部分的长度。如果新获得的定居点已经在城市内, 则要建造的城墙长度为零。

您现在想知道是否存在获取定居点的顺序, 以便每次移动要建造的墙的总长度不超过 m 。

输入

输入的第一行包含一个整数 T , 即测试用例的数量 ($1 \leq T \leq 5 \cdot 10^5$)。对于每个测试用例:

第一行包含两个整数 n 和 m , 即未占领定居点的数量以及一次移动可以建造的墙的最大长度 ($1 \leq n \leq 5 \cdot 10^5$ 和 $1 \leq m \leq 4 \cdot 10^9$)。

接下来的 n 行的第 i 行包含两个整数 x_i 和 y_i , 即聚落坐标 i ($1 \leq x_i, y_i \leq 10^9$)。

保证所有测试用例中的 n 之和不超过 $5 \cdot 10^5$ 。

输出

对于每个测试用例, 如果存在这样的顺序, 则输出包含单词“Yes”的行, 或者输出包含单词“No”的透明否则。



例子

<i>standard input</i>	<i>standard output</i>
3	Yes
3 6	No
1 1	Yes
4 1	
2 2	
4 9	
1 4	
2 3	
3 2	
4 1	
10 14	
10 8	
1 6	
2 5	
4 2	
5 5	
8 9	
2 7	
6 8	
6 5	
7 4	



问题 G. 反式

输入文件: 标准输入
输出文件: 标准
输出时间限制: 2 秒
内存限制: 512 MB

Bob 对 popcount 和一些奇怪的变换感兴趣。目前，他正在解决以下问题：

t 这是一个由 2^n 个整数 $a_0, a_1, a_2, \dots, a_{2^n-1}$ 组成的数组。任务是，对于每个 i ($0 \leq i \leq 2^n - 1$)，计算 b_i

$$b_i = \sum_{j=0}^{2^n-1} (\text{popcount}(i \text{ and } j) \bmod 2) \cdot a_j,$$

其中，“ $\text{popcount}(x)$ ” 表示 x 的二进制表示中 1 的数量，“and” 表示按位与运算。

尽管鲍勃很聪明，但他仍然不能快速解决问题。你能帮他计算所有的 b_i 吗？

输入

第一行包含一个整数 n ($1 \leq n \leq 20$)。第二行包含 2^n 个整数，描述数组 a ($1 \leq a_i \leq 10^9$)。

输出

打印一行包含 2^n 个整数，其中第 i 个整数是值 b_i 。

例子

<i>standard input</i>	<i>standard output</i>
2 1 2 3 4	0 6 7 5



问题 H. 盲盒

输入文件: 标准输入 输出文件: 标准
输出时间限制: 1 秒 内存限制: 512
MB

你是一家商店的老板。

该店推出了新的盲盒活动。每个盲盒内装有 n 张卡片，每张卡片上都写有一个正整数。每个盒子中的卡片的排序方式是，每 $i > 1$ ，第 i 张卡片上的数字大于或等于第 $(i-1)$ 张卡片上的数字。此外，每张卡片上的整数不超过 m 。

店里有满足上述条件的所有可能的盲盒，店里每两个盲盒都是不同的。当且仅当存在索引 i 使得两个盒子中第 i 张卡片上的数字不同时，两个盒子才被认为是不同的。

您以固定价格出售盲盒。购买并打开盲盒后，顾客会要求您返现，金额等于盲盒中 n 卡上数字的乘积。
请计算每个盲盒的最低价格，以确保售出所有盲盒后，您的净收入为非负值。

输入

输入的第一行包含两个整数 n 和 m : 每个盒子中的卡片数量以及卡片上的最大值 ($1 \leq n, m \leq 10^5$)。

输出

打印一个整数: 保证非负净收入的最低价格。价格可能是小数，但您必须以 998 244 353 为模输出该价格。
形式上，让最低价格为不可约分数 $\frac{x}{y}$ 。您必须打印 $x \cdot y^{-1} \bmod 998\,244\,353$ ，其中 y^{-1} 是一个整数，使得 $y \cdot y^{-1} \bmod 998\,244\,353 = 1$ 。

例子

<i>standard input</i>	<i>standard output</i>
2 2	332748120
5 5	499122514

笔记

第一个例子的解释:

共有三种不同的盲盒: (1, 1)、(1, 2) 和(2, 2)。返现金金额分别为 1、2、4。因此，最低价格应该是 $\frac{7}{3}$ 。第二个示例中的答案是 $\frac{42525}{126} = \frac{675}{2}$ 。

— —



问题一.EIP1559

输入文件: 标准输入 输出文件: 标准
输出时间限制: 3 秒 内存限制: 512
MB

您是一位狂热的以太坊研究员。最近，以太坊通过了一项决议，将交易的 Gas 费率从值 $gasPrice$ 更改为对 $(maxFee, maxPriorityFee)$ 。交易的确切 Gas 价格由 $gasPrice = \min(maxFee, maxPriorityFee + baseFee)$ 计算，而 $baseFee$ 是一个可以随时间变化的参数。

您维护一个动态的交易集合。有时，您想知道对于特定的 $baseFee$ ，集合中交易中最大的 $gasPrice$ 是多少。

具体来说，您需要维护一个支持以下三种操作的事务集合

S:

1. 将具有 Gas Rate $(maxFee, maxPriorityFee)$ 的交易添加到集合中。
2. 从集合中删除具有 Gas 费率 $(maxFee, maxPriorityFee)$ 的单个交易。保证至少有一笔交易满足条件。
3. 对于特定的 $baseFee$ ，当当前基本费用为 $baseFee$ 时，查找集合中 $gasPrice$ 的最大值。保证集合中至少有一笔交易。

输入

第一行包含一个整数 t ($0 \leq t \leq 10^6$) 表示操作数。对于以下 t 行，每行的第一个整数 $type$ 表示当前操作的类型。

如果 $type = 1$ ，则接下来的两个整数是 $maxFee$ 和 $maxPriorityFee$ 。您应该将具有 Gas Rate $(maxFee, maxPriorityFee)$ 的交易添加到集合中。

如果 $type = 2$ ，则接下来的两个整数是 $maxFee$ 和 $maxPriorityFee$ 。您应该从集合中删除具有 Gas Rate $(maxFee, maxPriorityFee)$ 的单个交易。

如果 $type = 3$ ，则下一个整数是 $baseFee$ 。当当前基本费用为 $baseFee$ 时，您应该输出集合中 $gasPrice$ 的最大值。

保证 $maxFee$ 、 $maxPriorityFee$ 和 $baseFee$ 的所有值都是 $[0, 10^6]$ 范围内的整数。

输出

对于每个 $type = 3$ 的操作，输出一行，其中包含表示当前基本费用为 $baseFee$ 时当前最大 $gasPrice$ 的整数。

例子

standard input	standard output
9	120000
1 200000 20000	140000
1 150000 40000	160000
1 120000 50000	130000
1 130000 30000	
3 80000	
3 100000	
3 140000	
2 150000 40000	
3 100000	



问题 J. 三个国家

输入文件: 标准输入 输出文件: 标准
输出时间限制: 1 秒 内存限制: 512
MB

今天, 您想要测量提瓦特的可达面积。

蒙德、璃月、闪电是提瓦特的三个国家。这些国家的领土可以分别视为三个圆圈 c_1 、 c_2 和 c_3 。有些圆圈可能重叠。

令 S_i 为 c_i 中的点集。Teyvat 的面积 S 被定义为 $S_1 \cup S_2 \cup S_3$ 中点的凸包

3。

形式上, S 是满足以下两个条件的最小点集:

- $S \supseteq S_1 \cup S_2 \cup S_3$,
- $\forall p_1, p_2 \in S, \forall \alpha \in [0, 1], \alpha p_1 + (1 - \alpha) p_2 \in S$ 。

您将获得圆圈 c_1 、 c_2 和 c_3 。你的任务是计算 S 的面积。

输入

第一行包含一个整数 t , 即测试用例的数量 ($1 \leq t \leq 10^4$)。

每个测试用例分三行给出。这些行的第 i 行包含三个整数: x 、 y 和 r , 它们是第 i 个圆的圆心和半径 ($1 \leq x, y, r \leq 100$) 的坐标。

输出

对于每个测试用例, 输出表示 S 面积的单个实数。

如果与评审团的答案相比, 您的答案的绝对或相对误差不超过 10^{-6} , 则您的答案将被视为正确。

例子

<i>standard input</i>	<i>standard output</i>
3	7.14159265359
1 1 1	8.79844690308
2 1 1	58923.76801932990
3 1 1	
1 1 1	
2 2 1	
3 3 1	
1 1 100	
85 27 100	
53 82 100	



问题 K. 包围猫

输入文件: 标准输入 输出文件: 标准
输出时间限制: 1 秒 内存限制: 512
MB

这是一个交互问题。

考虑平面上的六边形平铺, 如下图所示。

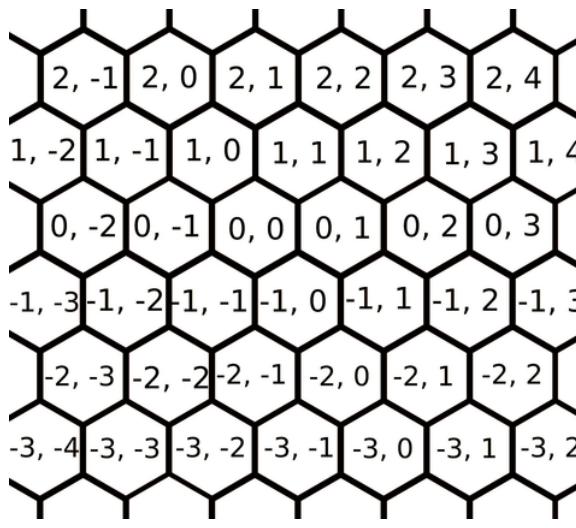
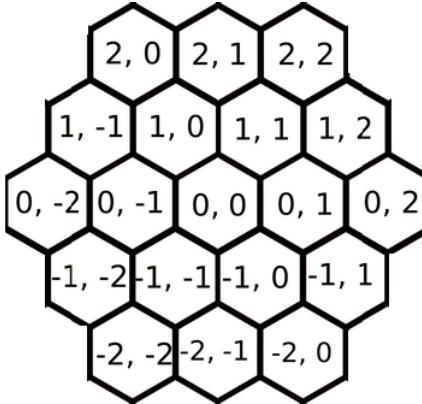


图 1: 飞机的样子。

你为你的猫建造的房子是一个边长为 $N = 10$ 的正六边形。它的六个角位于 $(9, 0)$, $(9, 9)$ 、 $(0, 9)$ 、 $(-9, 0)$ 、 $(-9, -9)$ 和 $(0, -9)$ 。

例如, 对于 $N = 3$, 房子看起来像这样:



现在猫想要逃跑。因此, 你需要在房子里放一些石头来围住猫的周围。
它无法逃脱。

最初, 猫位于 $(0, 0)$ 。每一秒, 你都可以在房子里选择一个位置并放置一块石头。请注意, 您不能在猫当前的位置放置一块石头。之后, 猫会选择一个与当前位置相邻且不包含岩石的位置, 并移动到那里。

一旦猫到达房子的边界, 它就会逃脱并获胜。另一方面, 如果猫无法选择有效的举动, 你就赢了。尝试包围猫, 使其无法逃脱。

交互协议

使用标准输入读取

猫的位置。使用标准输出来写

列出地点



你放置岩石。每个位置都在一行中给出，其中包含两个空格分隔的整数：位置的坐标。

最初，猫位于 $(0, 0)$ ，并为您提供 “0 0” 作为输入。您通过打印第一块石头的位置来响应，然后读取猫移动的下一个位置，依此类推。

如果您打印出无效位置（不在房子内，或当前包含猫），请终止程序以获得“错误答案”判决。允许将岩石放置在已经包含岩石的位置，但不会产生任何额外效果。如果猫到达了房子的边界，你就输了：终止你的程序以获得“错误答案”的判决。如果猫没有有效的动作，你就赢了：不必读取猫的下一个位置，只需优雅地终止你的程序即可。

不要忘记打印换行符并在打印每个位置后刷新输出：否则，您可能会得到“超出空闲限制”的结论。

例子

<i>standard input</i>	<i>standard output</i>
0 0	1 1
1 0	0 1
0 0	-1 0
1 0	-1 -1
0 0	0 -1
1 0	9 9
0 0	1 0