

第三届环球杯



第 6 赛段奥西耶克

2024 年 8 月 10-11 日

该问题集应包含 12 个问题，共 19 页。

根据



奥西耶克编程竞技营



问题 A.同位数组

10-11 日

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 256 兆字节

给定两个整数 s 和 x ，求任意一个最短数组，使其元素之和为 s ，且所有元素与 x 共素。
如果两个整数相除的唯一正整数是 1，那么这两个整数是共质数。

输入

唯一的一行包含整数 s 和 x ($2 \leq s, x \leq 10^9$)。

输出

如果没有满足条件的数组，则打印单个整数-1。
否则，第一行应包含一个整数 n ($1 \leq n \leq 10^6$) --数组的长度。下一行应包含 n 个空格分隔的整数--数组本身。数组元素的绝对值不应超过 10^9 。
如果有多个可能的答案，请打印任意一个。我们已经证明，如果存在一个解，那么就存在一个满足上述约束条件的解。

实例

标准输入	标准输出
9 6	3 -7 -7 23
14 34	2 83 -69



问题 B. 方形定位器

10-11 日

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：256 兆字节

有一个正方形 $ABCD$ ，其顶点的坐标是整数， A 在 y 轴的正方向上。
给你 AO^2, BO^2, CO^2, DO^2 的平方距离，其中 $O(0, 0)$ 是原点。请找出正方形的顶点。

输入

每个测试的唯一一行包含四个整数 AO^2, BO^2, CO^2, DO^2 ($1 \leq AO^2, BO^2, CO^2, DO^2 \leq 10^{18}$) --即正方形顶点到原点距离的平方。

输出

输出一行，包含七个空格分隔的整数 $A_y, B_x, B_y, C_x, C_y, D_x, D_y$ ，代表正方形顶点的坐标 $A(0, A_y), B(B_x, B_y), C(C_x, C_y), D(D_x, D_y)$ 。
输入的内容就是这样一个整数。如果有多个可能的答案，则打印其中任何一个。

示例

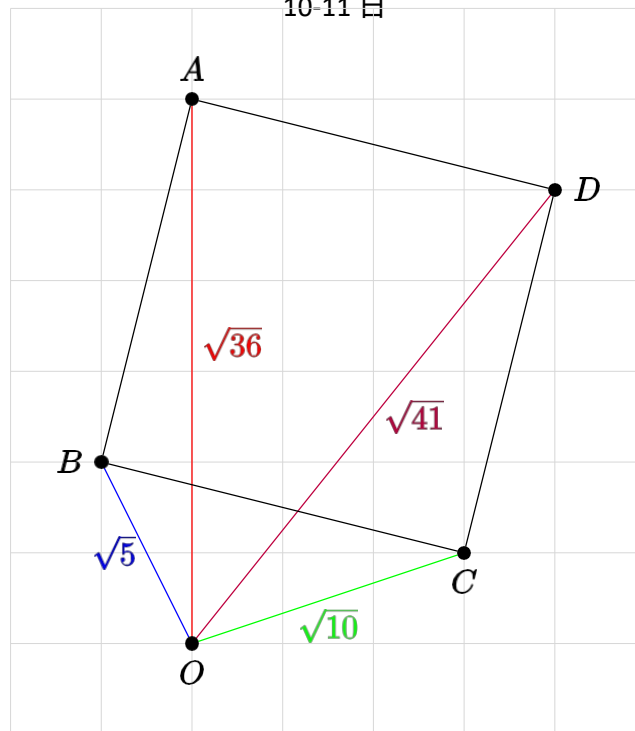
标准输入	标准输出
36 5 10 41	6 -1 2 3 1 4 5

备注

第一个测试案例如下图所示。



10-11 日





问题 C. -is-this-bitset-

10-11 日

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 256 兆字节

注意内存限制较低！

给你一棵有 n 个节点的二叉树，根节点为 1。这意味着每个节点最多有 2 个孩子您还将得到两个包含 n 个整数的数组 a 和 b

节点 i 的子集问题定义为能否从节点 i 的祖先及其自身中取一个子集 S ，使得 $\sum_{j \in S} (a_j) = b_i$

在一次操作中，你选择两个整数 i 和 x ($1 \leq i \leq n$)、
 $0 \leq x \leq 2 \cdot 10^6$)，并设置 $a_i := x$ 。

完成这些操作后，求解每个节点 i 的子集和问题，并以位字符串的形式输出结果。

输入

接下来的 $n - 1$ 行包含两个整数 u 和 v ($1 \leq u, v \leq n, u \neq v$) --节点 u 和 v 由一条边连接。

保证这些边构成一棵根植于节点 1 的二叉树

下一行包含 n 个整数 a_1, a_2, \dots, a_n ($0 \leq a_i \leq 2 \cdot 10^6$) -数组 a 。

输入的最后一行包含 n 个整数 b_1, b_2, \dots, b_n ($0 \leq b_i \leq 2 \cdot 10^6$) --数组 b

输出

输出 n 个整数 a'_1, a'_2, \dots, a'_n ($0 \leq a'_i \leq 2 \cdot 10^6$) - 新数组 a' ，即经过对其进行了操作。

在下一行中，输出长度为 n 的位字符串，如果节点 i 上的子集问题为 1，则在位置 i 上输出 1 可以求解，否则为 0。

实例

标准输入	标准输出
5 2 1 1 3 3 4 5 4 1 3 11 12 6 0 5 12 13 18	1 3 11 12 0 10110
1 2000000	2000000 1



2000000

10-11 日

备注

在示例输出中，我们决定将数组 a 的最后一个数字改为 0。



问题 D. 循环游戏

10-11 日

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：256 兆字节

有一块棋盘，上面有 $n \times m$ 个白色方格。泽罗在这个方格上玩单人游戏。每轮他都会给一个方格涂上黑色。如果两个方格共享一条边，那么它们就是相邻的。这些方格和这些相邻关系构成了一个平面图。当黑色方格形成一个内部不空的简单循环时，游戏结束。在方格中心之间用直线画出图形的边，就能得到平面图形。如果不在循环上的正方形位于循环边所构成的多边形的内部，则该正方形位于循环的内部。泽罗希望下完棋后能在棋盘上画出一幅漂亮的图画，因此他已经想好了要下的棋步。如果某步棋恰好会结束对局，他就不会下这步棋。在输入中，有 k 个不同的位置 (r_i, c_i) ，即耶罗计划好的棋步。

你的任务是计算出每一步棋是否会结束对局，如果第 i 步棋最终下完，则打印出带有 1 的位字符串，如果这步棋会结束对局但没有下完，则打印出带有 0 的位字符串。

输入

输入的第一行包含三个整数 n, m, k ($1 \leq n - m \leq 300\,000$, $1 \leq k \leq n - m$)。接下来的 k 行分别包含两个整数 r_i, c_i ($1 \leq r_i \leq n$, $1 \leq c_i \leq m$)，它们是泽罗计划下的第 i 步棋的行和列。

输出

输出一行长度为 k 的字符串，如果要下第 i 步棋，则第 i 个位置上的字符串为 1；如果这步棋最终会输，因此不会下，则字符串为 0。

实例

标准输入	标准输出
4 3 7 2 1 2 2 2 3 3 1 3 2 4 1 4 2	1111111



3 3 8	11111110
1 1	10-11 日
1 2	
1 3	
2 3	
3 3	
3 2	
3 1	
2 1	



问题 E. 平方和

10-11 日

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

256 兆字节

给定多项式 $A(x) = a_0 + \dots + a_n x^n$ ，系数为整数， m 为整数。考虑多变量多项式 $D(x_1, \dots, x_m)$ ，其定义为

$$D(x_1, \dots, x_m) = \prod_{i=1}^m A(x_i) \prod_{j=1}^{i-1} (x_i - x_j).$$

设 s 是 $D(x_1, \dots, x_m)$ 所有系数的平方和。求 $s \bmod 10^9 + 7$ 。

输入

唯一的一行输入包含两个整数 n ($0 \leq n \leq 500$) 和 m ($0 \leq m \leq 10^9$)。第二

行包含 $n + 1$ 个整数 a_0, \dots, a_n ($0 \leq a_i < 10^9 + 7$)。

可以保证 $a_n \neq 0$ 和 $a_0 \neq 0$ 。

输出

打印一个整数，即 $s \bmod 10^9 + 7$ 的值。

实例

标准输入	标准输出
2 0 1 2 3	1
2 1 1 2 3	14
2 2 1 2 3	264

备注

对于 $A(x) = 1 + 2x + 3x^2$ 和 $m = 2$ ，我们有

$$D(x_1, x_2) = \underset{0}{-9x_1^2x_2^2} - \underset{12}{6x_1^3x_2} - \underset{1}{3x_1^3} + \underset{12}{9x_1^2x_2^3} - \underset{1}{x_1^2x_2^2} - \underset{1}{2x_1^2} + \underset{2}{6x_1x_1^3} + \underset{2}{x_1x_1^2} - \underset{2}{x_1} + \underset{2}{3x_1^3} + \underset{2}{2x_1^2} + x_2$$

请注意，当 $m = 0$ 时， $D = 1$ ；当 $m = 1$ 时， $D(x_1) = 1 + 2x_1 + 3x_1^2$ 。



问题 F. 交替循环

10-11 日

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：256 兆字节

给你平面上的 n 个点，其中没有 3 个点是相邻的。您可以选择这些点的某个非空子集，并为这个子集选择一个顺序。如果你选择的有序子集中的点是 p_0, p_1, \dots, p_{k-1} 我们希望在 $1 \leq i \leq k+1$ 的条件下， $\angle p_{i-1} p_i p_{i+1}$ 的角按 i 的递增顺序顺时针和逆时针交替排列。这样有序的点子集称为交替循环。

您必须找到其中点数最少的交替循环，并将其打印输出，否则就报告不存在这样的循环。

输入

第一行输入包含一个整数 n ($1 \leq n \leq 200\,000$)，即输入中的点数。
接下来的 n 行中，每一行都包含一个点的描述。每行包含两个整数 x 和 y ，($0 \leq x, y \leq 10^9$) -- 即点的坐标。
保证点是不同的，也保证没有 3 个点是相邻的。

输出

如果没有交替循环，则输出-1。否则输出 k ，即循环中的点数。以下各行依次输出循环中的点。
。在下面的 k 行中，分别打印两个整数 x 和 y ($0 \leq x, y \leq 10^9$) --交替循环中一个点的坐标。
这些点应该是输入点的子集，它们应该按照输入的顺序形成交替循环。
如果有多个解决方案都能达到 k 的最小值，那么其中任何一个解决方案都会被接受。

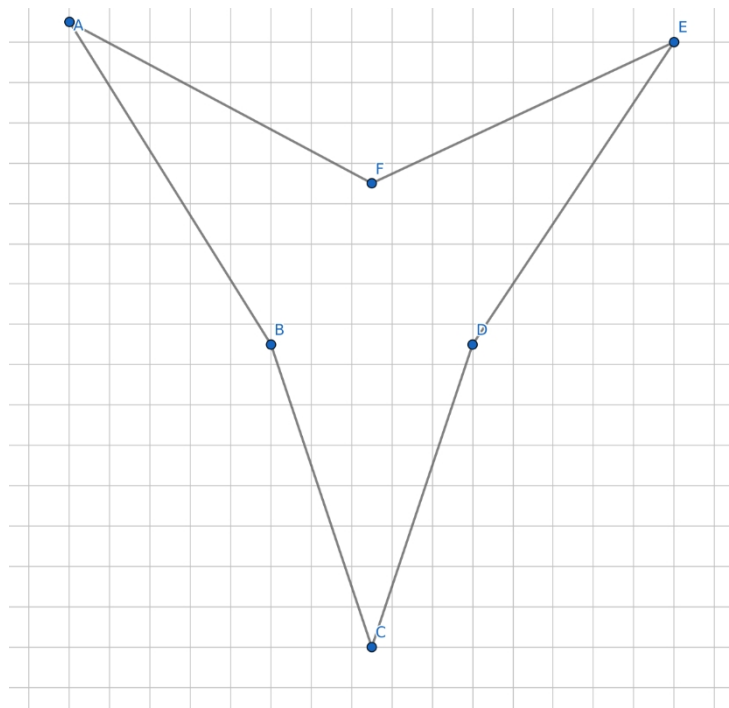
实例

标准输入	标准输出
6 10 15 20 15 15 23 0 31 15 0 30 30	6 0 31 10 15 15 0 20 15 30 30 15 23
4 0 0 0 1	-1



1 0	10-11 日
1 1	

备注



样本 1 的示意图，用直线段绘制了长度为 6 的可能交替循环。



问题 G. 接触草地

10-11 日

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	32 兆字节

请注意不寻常的内存限制。

Adamant 是一个以撰写数学和编程教育博文而闻名的人。有一天，他终于决定，他已经写了太多的博客，是时候到外面去摸摸草了。

Adamant 有一块草坪，上面长着 n 株小草。我们可以在二维平面上画出草坪和草：地面是 $y = 0$ 的直线，草垂直向上生长。每棵草都可以用 x 和 y 这两个数字来描述，也就是说，它可以看作是从 $(x, 0)$ 到 (x, y) 的一条垂直线段。

为了摸到草，阿达曼特将移动他的手 m 次。每次他都会将手从坐标 (x_1, y_1) 沿直线移动到坐标 (x_2, y_2) 处。如果线段 $(x_1, y_1) - (x_2, y_2)$ 接触到由第 i 根草定义的线段（或其端点），我们就说他的手碰到了第 i 根草。

您的任务是确定 Adamant 每次移动手时，是否会碰到任何草。如果答案是肯定的，你还需要找出他将会碰到的草的指数。

输入

第一行包含一个整数 n ($1 \leq n \leq 8 \times 10^5$)。

接下来 n 行中的第 i 行包含两个整数 x, y ($1 \leq x, y \leq 10^9$)，用于描述索引为 i 的草。

下一行包含一个整数 m ($1 \leq m \leq 3 \times 10^5$)。

接下来的 m 行中，每一行都包含四个整数 x_1, y_1, x_2, y_2 ($1 \leq x_1, y_1, x_2, y_2 \leq 10^9$)，描述一个手部动作。保证 x_1 和 x_2 都不同于任何草的 x 坐标。

输出

手每移动一次，打印一行：所触及的任何草的索引。如果有多个答案，则打印任意一个。如果没有碰到草，则打印-1。

示例

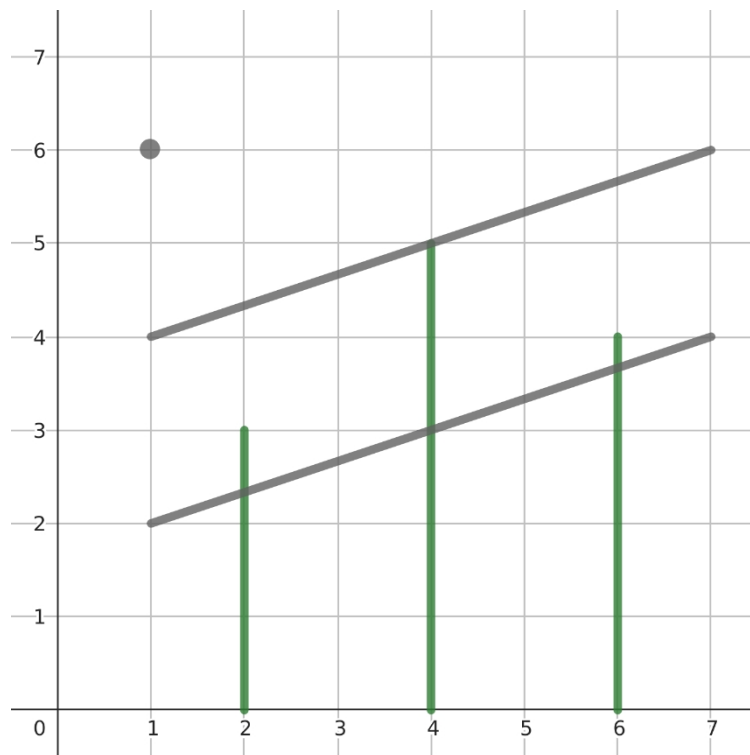
标准输入	标准输出
3	3
2 3	1
6 4	-1



4 5 3 1 4 7 6 7 4 1 2 1 6 1 6	10-11 日
-------------------------------------------	---------

备注

样品的可视化：





问题 H. 游戏设计

10-11 日

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：256 兆字节

你是一名游戏设计师，正在设计一款新的视频游戏 "杰出迷人的平台挑战"。游戏由 n 个关卡组成，关卡编号为 $1 \dots n$ ，玩家必须按顺序完成这些关卡。除了正常的进程外，这些关卡还通过 n 个单向翘曲门户相连。贵公司的一个不同团队已经完成了每个关卡的设计。他们在每个关卡都设置了一个曲速入口和一个曲速出口。您的任务是将每个入口传送门连接到不同楼层的出口传送门，这样每个出口传送门也只连接到一个入口传送门。

不过，还有一个额外的限制：玩家不能在游戏中跳过前面的关卡。也就是说，玩家不能进入一个传送门，从后面关卡的传送门出来，然后继续玩后面的关卡。为了做到这一点，关卡设计者将一些出口传送门设置在与关卡其他部分隔绝的位置。也就是说：如果 u 层的入口入口通向 v 层 $> u$ 层的出口入口，那么 v 层的出口入口必须位于一个孤立的位置。

你已经编写了一个程序，来研究将每个入口连接到出口的所有可行方法，以便测量预测的受众参与度。这个程序已经运行了一段时间，你的老板正在生气，你想知道这个程序需要多长时间。因此，请计算将每个入口连接到出口的允许方式数。打印答案：998 244 353。

输入

第一行包含一个整数 t ($1 \leq t \leq 1000$) --测试用例数。
每个测试用例由长度为 n ($2 \leq n \leq 5000$) 的二进制字符串 s 组成。如果第 i 层的出口门户位于隔离位置，则 s 的第 i 个字符为 1，否则为 0。
保证所有测试用例的 n 之和不超过 5000。

输出

对每个测试用例，另起一行打印答案的模数 998 244 353。

示例

标准输入	标准输出
4	3
0101	0
1010010010001010	44
11111	393298077
10100100011000010010101001001001	



备注

10-11 日

在第一个测试示例中，有效配置为[2, 1, 4, 3]、[2, 4, 1, 3]和[4, 1, 2, 3]，其中阵列中的第 i 个位置是出口入口与第 i 层入口相连的位置。

在第二个测试示例中，没有入口门户可以连接到最后一层的出口门户。



问题 I. 几何黑客

10-11 日

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：256 兆字节

Jeroen 制作了一种新算法，用于检查一个点是否在多边形内：

给定一个简单多边形 $p_0, p_1 \dots p_{n-1}$ 和一个查询点 q ，他的算法如下：

- 初始化 $ans := 0$
- 从 q 沿 x 正方向画一条射线。
- 对于所有 $0 \leq i < n$ ：检查闭合线段 $p_i p_{(i+1) \bmod n}$ 是否与射线相交，如果是，则递增答案。
- 对于所有 $0 \leq i < n$ ，如果 p_i 位于射线上，则在答案上加 1。
- 如果 ans 是奇数，算法会说在内部，否则在外部。

他坚信自己的算法是正确的，但你必须说服他。为了羞辱他，你要寻找点 $(0, 0)$ 严格位于多边形内部（因此不在边界上）的多边形，但他的算法会说点在外部。考虑所有顶点在整数坐标上且这种情况成立的简单多边形，并按面积对这个无限列表进行排序，任意打破平局。给定输入中的整数 $k \leq 10^3$ ，输出列表中排在前面的 k 个多边形。如果存在多个可能的答案，可以打印其中任何一个。

请注意，如果 a 的点列表是 b 的点的循环旋转，则两个多边形 a 和 b 被视为相同的多边形。请注意，虽然在多边形的一条边上多放一个整数点会产生相同的形状，但这将被视为不同的多边形。

输入

唯一的输入行包含整数 k ($1 \leq k \leq 1000$)，即应输出的多边形数量。

输出

按面积大小排序，输出泽罗算法失败的前 k 个多边形的描述：在描述的第一行，输出 n ，即多边形的顶点数。

在接下来的 n 行中，每行输出 2 个整数 x 和 y ($|x|, |y| \leq 10^9$)，即多边形顶点的坐标。

在问题的约束条件下，可以证明如果前 k 个有效多边形的最小面积为 $A_1 \leq A_2 \leq \dots \leq A_k$ ，那么确保坐标绝对值不超过 10^9 的附加约束条件不会改变这个面积列表。



请注意，输出中的多边形必须是简单多边形40-11 日



示例

10-11 日

标准输入	标准输出
2	4 -1 0 0 1 1 0 0 -1 3 0 -4 3 5 -3 5

备注

样本输入的输出实际上是错误的。这只是为了显示输出格式。这两个多边形的内部都有一个点 $(0, 0)$ ，但耶罗的算法并没有失败，它们的面积并不能保证是可能的最小的两个面积。



问题 J. 非交互式 Nim

10-11 日

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

256 兆字节

Nim 是一种经典的策略游戏，由两名玩家进行。共有 n 堆棋子，其中第 i 堆包含 a_i 个棋子。玩家轮流下棋。在每一轮中，棋手必须选取一堆有正数个石子的棋子，并从中取出正数个石子。不能移动的玩家输。

蓝色怪兽希望在 "普通与常见问题集" 中添加一个交互式尼姆问题。在这个问题中，你的程序必须与一个总是最佳下法的对手进行尼姆对弈。然而，蓝怪懒得学习如何创建交互式问题。因此，我们要求你只在对手只有一步最优棋的情况下下棋。

你会得到一个 a_1, a_2, \dots, a_n - 一个 Nim 实例。如果两个玩家都以最佳方式下棋，那么保证第一个玩家会输。您将作为第一位下棋者，您的对手将作为第二位下棋者。您的对手总是会下出最佳棋步。您必须以这样的方式下棋：在您的每一步棋之后，您的对手只能下一步棋，这样如果两位棋手在这一步之后都以最佳方式下棋，您就会输。

打印一连串这样的棋步，或者宣布这是不可能的。请注意，由于对手总是下出最佳棋步，因此你很清楚对手会下出什么棋。您不必尽量减少棋步序列的长度。

输入

第一行包含一个整数 t ($1 \leq t \leq 5 \cdot 10^4$) -- 测试用例数。随后是 t 个测试用例。每个测试用例的描述如下。

测试用例的第一行包含一个整数 n ($2 \leq n \leq 10^5$)。第二行包含 n 个整数 a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^{18}$)。可以保证的是，如果两个棋手都以最佳状态下棋，第一个棋手将会输。

保证所有测试用例的 n 之和不超过 10^5 。

输出

对于每个测试用例，打印答案如下。

如果棋局不可能总是让对手只有一步最佳棋步，则打印 -1。否则，在第一行打印一个整数 k ($1 \leq k \leq 100$) -- 棋步数。在接下来的 k 行中，每行打印两个整数 p ($1 \leq p \leq n$) 和 x ，表示将从第 p 个棋子堆中取出 x 个棋子。

可以证明，在这一问题的约束条件下，如果有可能下得让对手始终只有一步最优棋，那么就有



可能做到这一点，并在 100 步内输掉对局。 10-11 日

示例

标准输入	标准输出
2	4
4	3 2
4 2 7 1	1 2
4	3 3
1 1 1 1	4 1
	-1



备注

10-11 日

在第一个测试实例中，对手被迫下棋 $(2, 2)$ 、 $(3, 2)$ 、 $(1, 1)$ 和 $(1, 1)$ 。对局过程如下

搬家后	对手移动后
4, 2, 5, 1	
	4, 0, 5, 1
2, 0, 5, 1	
	2, 0, 3, 1
2, 0, 0, 1	
	1, 0, 0, 1
1, 0, 0, 0	
	0, 0, 0, 0

在第二个例子的测试中，无论你下哪一步棋，都会剩下三个非空的棋子堆，每个堆都有一颗棋子。显然，对手的所有选择都是等价的。



问题 K. 绳子和钉子

10-11 日

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：256 兆字节

在一块木板上有 n 个钉子，钉子的坐标为整数。您有一根绳子，它将所有钉子都包住，拉紧后形成一个最小的环，将各点都包住。您必须按照以下规则逐个取出钉子：

首先，将绳子拉紧剩余的钉子。然后，你可以选择绳子绷紧的任何一个钉子（这样绳子就会接触到这个钉子，并形成一个小于 180 度的角度），然后把它取下来。重复这个过程，直到只剩下一个点。

检查是否可以重复拔除钉子，直到只剩下一个钉子，如果可以，请给出拔除的顺序。

输入

第一行输入包含一个整数 n ($1 \leq n \leq 200\,000$)，即输入中的点数。

接下来的 n 行中，每一行都包含一个钉子的描述。每行包含两个整数 x 和 y ， ($0 \leq x, y \leq 10^9$) -- 即钉子在木板上的坐标。

保证没有两颗钉子在木板上占据完全相同的位置。

输出

如果可以根据规则移除除一个钉子以外的所有钉子，则输出 "是"；否则，打印 "否"。如果答案是 "是"，则在下面几行输出移除结果。

在下面的 $n - 1$ 行中，分别打印两个整数 x 和 y ($0 \leq x, y \leq 10^9$) --即当前要移除的钉子的坐标。

如果有多个解决方案，则接受任何解决方案。

实例

标准输入	标准输出
3 1 1 2 4 3 1	是 1 1 2 4
1 1000000000 0	是



问题 L. 吃到饱

10-11 日

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
	256 兆字节

这是一个互动问题。

在全自助传送带餐厅，餐点被放在一条移动的传送带上。顾客坐在传送带旁边，餐点从旁边滑过。你可以在餐点从你身边滑过时拿走任何一份餐点，而且可以重复多次。这很划算，但也很容易让人暴饮暴食，事后感到不适。

具体来说，您坐下后，1、2、....., n 餐会滑过。看到第 i 餐后，您判断其热量值为 i 。

在一家这样的餐厅里，有两种类型的座位：

- *面向传送带相反的方向*。在这种情况下，你可以看到所有将来会到达你面前的物品，从而可以相应地计划你的选择（假设餐厅里没有其他顾客）。
- *面向传送带的移动方向*。在这种情况下，你一次只能看到一种食物：当第 i 种食物靠近你时，你必须决定要不要吃，而不知道它后面的食物的热量值是多少。一旦饭菜 $i + 1$ 滑过，饭菜 i 就已经够不着了。

因此，面对传送带的方向更难优化你的午餐。不过，你倒是想出了一个不道德的小窍门：你可以悄悄地把自己挑好的餐点放到另一位顾客的桌子上，这样就可以把它扔掉了。

为了防止暴饮暴食，您为自己制定了以下规则。

- 您将面对传送带的运动方向。
- 一旦你开始进食，就不能再从传送带上取走任何新的食物。
- 在任何时候，您餐桌上饭菜的总热量不得超过 1000 卡路里。

现在你遇到了相反的问题--你担心如果你遵守所有规则，离开餐厅时就会饿肚子。你决定，只要你所吃的饭菜的总热量至少是你朝相反方向但仍遵守其他规则所能得到的热量的 60%，你就会感到高兴。

更正式地说，设 x 是 a_1, a_2, \dots 的子序列中不超过 1000 的最大可能和， a_n 不超过 1000 的最大可能之和。如果三餐的总热量至少是 $0.6x$ ，你就会很高兴。



实施一种策略，确保你永远快乐。交互器是**自适应的**，也就是说，热量值的顺序并不一定是事先决定好的：它可能取决于你的程序所做的选择。

输入

第一行包含一个整数 t ($1 \leq t \leq 10^4$) -- 测试用例数。

互动协议

你的程序与评委程序之间的交互从读取一个整数 n 开始
($1 \leq n \leq 10^4$) - 将滑过的餐数。以下情况将发生 n 次：



- 从输入中读取一个整数 a_i ($0 \leq a_i \leq 1000$) -- 第 i 餐的热量值。
- 打印一行格式为 $k\ t_1\ t_2\ \dots\ t_k$, ($0 \leq k \leq i, 1 \leq t_j \leq i$) 的表格，表示您要放弃索引为 t_1, t_2, \dots 的餐点。所有的餐点当时都必须在您的餐桌上。
- 打印一行，其中包含单词 "TAKE" (如果您想在餐桌上添加 i 餐) 或 "IGNORE" (如果您想将其留在传送带上)。无论哪种情况，打印的单词都必须去掉引号。

经过 n 次迭代后，表格中的卡路里值总和最多必须为 $0.6x$ ，其中 x 的定义如上。

保证所有测试用例的 n 之和最多为 10^4 。

如果您的程序在任何时候进行了无效查询（例如，打印了不符合输入规范的行，表中的总值一次超过 1000，试图丢弃表中没有的项目，或者测试用例在您不满意的情况下结束），交互器将立即终止。如果您的程序在此之后继续读取输入，可能会收到任意判决，因为它将继续从一个关闭的流中读取数据。为了避免这种情况，请始终检查输入流是否仍处于打开状态，即不要写入

```
int ai;
cin >> ai;
;
```

写道

```
int ai;
if (!(cin >> ai)) exit(0);
;
```

以在交互程序终止时立即终止程序。这样，如果您的查询无效，就会收到 "错误答案"。

打印查询后，不要忘记输出行尾并刷新输出。为此，请使用

- `fflush(stdout)` 或 C++ 中的 `cout.flush()`;
- 在 Python 中使用 `stdout.flush()`。

示例

标准输入	标准输出
1	
5	
10	0
	采取
13	
	0
	采取
450	
	0



585	采取 10-11 日
465	2 1 3 采取 0 忽略



备注

10-11 日

示例部分展示了您的程序与法官之间的一种可能互动。在第三项之后，您的桌子上还有 1、2、3 项，热量值分别为 10、13 和 450。当第四件物品到来时，如果您想拿走它，就必须放弃一些物品，因为 $10 + 13 + 450 + 585 = 1058 > 1000$ 。在本例中，您的程序决定舍弃第 1 项和第 3 项，因此之后正好有 598 项。你也可以忽略最后一项。

整个过程结束后，桌上的总热量值为 598。如果你面对的是相反的方向，你可能会有多达 $10 + 13 + 450 + 465 = 938$ 。由于 $\frac{598}{938} \approx 0.637 > 0.6$ ，所以这是有效的。

解决方案