DeepL

Subscribe to DeepL Pro to translate larger documents.
Visit www.DeepL.com/pro for more information.

问题 A. 接管

输入文件： *标准输入*
输出文件： *标准输出*
时间限制： 2 秒 2 秒 内存限制
512 兆字节

一家新的 IT 公司最近在该市成立了办事处！他们的总部位于点 (0，0)， 园区是一个矩形，四角分别位于 (0, 0) 和 (1，1)。园区非常小，还没有任何设施。然而，附近有许多设施。为什么不扩大校园，把它们都放在校园里呢？

这些设施将被逐一捕获并添加到校园中。控制每个设施后，公司必须重建园区周围的围栏。围墙应是边与坐标轴平行的最小矩形，包含总部和所有占领的设施。

可以重复使用以前围栏的材料，但有时围栏的总长度可能会增加。在这种情况下，有必要购买一些新材料。如果捕获前栅栏的长度是 a 米，捕获后变成了 b 米，那么就需要购买 $b-a$ 单位的材料。

很难证明需要购买大量材料。找出占领设施的顺序，使占领后围栏长度的最大增加量最小。

建造初始围栏（大小为 4）所需的材料不视为增加，因为它最初就存在。

输入

第一行输入的是整数 n（$1\leq n\leq 3\cdot 10^5$），即设施数量。接下来的 n 行中，每一行都有两个整数 x_i和 y_i（$1\leq x_i, y_i\leq 10^9$），即*第 i 个*设施的坐标。

允许两个设施位于同一位置，也允许任何设施最初已在园区内。

输出

打印从 1 到 n *的*数字排列：设施捕捉的顺序。该顺序应尽量减少捕捉后栅栏长度的最大增长。设施从 1 到 n 按输入的顺序编号。

如果有多个答案，请打印任意一个。

示例

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

问题 B. 不公平的纸牌

输入文件： *标准输入*
输出文件： *标准输出*
时间限制： 2 秒 2 秒 内存限制
512 兆字节

在另一款纸牌游戏中，有几种类型的纸牌和一副 30 张的纸牌。对于每种纸牌类型，牌组中要么只有一张该类型的纸牌，要么有两张这样的纸牌。最初，这副扑克牌是洗好的。然后，从一副牌中一张一张地随机抽取，直到抽空为止。

在玩了 100 000 局游戏之后，您发现庄家抽牌不公平。他不是随机抽取下一张牌，而是遵循某种奇怪的算法。*第* i 张牌的类型被赋予一个权重 $X_(i)$（$0< X_i\leq 1$）。每轮抽出的下一张牌是*第* i 种类型的概率是 $c_{(i)}\,X_{(i)}/S$，其中 c_i是剩余的*第* i 种类型的牌的数量，$S=\sum_j c_{(j)}\,X_{(j)}$是所有剩余牌的权重之和。

为了为下一次游戏做好准备，您希望能够预测庄家的行动。幸运的是，你还记得所有 100 100 局游戏中的抽牌顺序。根据这些信息，找出分配给每种牌的权重。

输入

在第一行中，有两个整数 m 和 n（$m=100\,000$，$1\leq n\leq 30$）、游戏次数和牌型次数。

下一行有 n 个整数 $a_1, \ldots, a_n$($1\leq a_1\leq 2$、$\sum_{i=1}^{n} a_i=30$），表示*第* i 张牌有多少张。
类型的牌有多少张。

接下来*的*每一行都包含一局游戏的日志。对数日志由 30 个数字组成：按照抽牌顺序排列的牌的类型。每 $1\leq i\leq n$，数字 i 在日志中出现的次数*为* a_i。

输出

打印 n 个数字 $W_1, \ldots, W_n$($10^{-300}< W_i\leq 1$)，即预测的纸牌类型权重。对于每一对类型 i 和 j，如果 $X_i\leq X_{(j)}$，下面的条件都成立，那么答案将被认为是正确的：

$$\left|\frac{X_i}{X_i+X_j}-\frac{W_i}{W_i+W_j}\right|<0.02$$

注

小输入示例（注意，测试集中不会出现该示例，因为 m 始终为 100 000）：

```
4 3
2 1 1
1 1 2 3
1 2 1 3
1 2 3 1
2 1 1 3
```

问题 C. 多样化歌唱

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 兆字节

在现代社会，多样性是人们最关心的问题，尤其是语言多样性。因此，即将举行的选秀节目将使用多种语言。然而，如何让节目既多元化又不枯燥乏味，却是一个棘手的问题。

有 n 名歌手参加演出，要演唱 m 首歌曲。每位歌手都有多首歌曲，其中一些可能用不同的语言演唱。一个节目要想完整，每个参与者至少要唱一首歌，而且每首歌至少要唱一次。要使节目不枯燥，每位参与者使用每种语言的次数不应超过一次，每首歌用每种语言演唱的次数也不应超过一次。

根据每位歌手的演唱曲目，制作一个完整而不枯燥的节目，或者确定这是不可能的。

输入

第一行输入三个整数 n、m、k：分别表示歌手、歌曲和可能的表演的数量（$1\le n$，$m\le 1000$，$1\le k\le 10\,000$）。

接下来的 k 行描述曲目。*第* i 行包含三个整数 $p_i, s_{(i)}, l_i$（$1\le p_i\le n$, $1\le s(_i)\le m$, $1\le l_i\le k$），表示参与者 p_i可以用语言 $l_{(i)}$演唱歌曲 $s_{(i)}$。

输出

如果无法编制一个完整而不枯燥的程序，则打印一个数字"-1"（不带引号）。否则，在第一行打印整数 t：要演奏的歌曲数量。第二行打印 t

个介于 1 和 k 之间的不同整数：要包含在节目中的曲目条目。曲目编号从
1 的顺序编号。曲目条目可以任意顺序打印。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
2 2 4 1 1 1 1 2 1 1 2 2 2 2 2	2 1 4
2 3 5 1 1 1 1 2 1 2 2 2 2 3 2 2 2 3	3 1 4 5
2 3 4 1 1 1 1 2 1 2 2 2 2 3 2	-1

问题 D. 自选尼姆

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 兆字节

爱丽丝和鲍勃喜欢玩尼姆游戏（如果您不记得游戏规则，请参阅注释部分）。他们玩了很多次，以至于一眼就学会了如何决定胜负：如果堆中有 $a_1, \ldots$.如果堆中有 $a_{(1)}, \ldots a_{(n)}$ 个棋子，那么只有当且仅当比特 xor $a_1 \oplus \ldots \oplus a_n$是非零时，第一个玩家才会获胜。

他们听说，在一些网络游戏中，玩家会在游戏前选择自己的角色，这就增加了一个战略层面。为什么不在 Nim 中这样做呢？

他们提出了以下版本。爱丽丝和鲍勃各有几个装有堆的盒子。在第一阶段，他们从每个盒子中选出一个堆。在第二阶段，爱丽丝从这些堆中选择一些非空子集，然后在选定的堆上开始常规的尼姆游戏，由鲍勃先移动。

鲍勃已经知道爱丽丝选了哪些堆。请帮助他进行选择，这样无论爱丽丝在第二阶段选择了哪些堆，他都能赢得游戏。

输入

第一行是一个整数 n（$0 \leq n \leq 60$），即爱丽丝选择的堆数。

如果 $n > 0$，则下一行有 n 个整数：这些堆的大小。否则，这一行省略。下一行是一个数字 m（$1 \leq m \leq 60$），即鲍勃的盒子数

量。

接下来的 m 行中，每一行都包含一个盒子的描述。每段描述都以一个数字 k_i开始（$1 \leq k_i \leq 5000$），即盒子中的堆数。然后是 $k_{(i)}$ 个数字，表示这些堆的大小。

每个堆的大小介于 1 和 $2^{60}-1$ 之间。鲍勃盒子中的堆总数不超过 5000。

输出

如果鲍勃无法获胜（也就是说，无论他选择什么，爱丽丝都可以做出这样的选择，从而导致尼姆位置输掉），则打印"-1"（不带引号）。否则，打印 m 个整数：鲍勃应从他的盒子中挑选的堆的大小，顺序与输入中给出的盒子顺序相同。

示例

标准输入	标准输出
2 1 2 2 2 1 2 3 1 2 3	-1
1 5 2 3 1 2 3 4 4 5 6 7	1 6

注意事项

在 "尼姆 "游戏中，有几堆棋子。在每个回合中，棋手选择任意一堆，并从中取走一定正数的棋子。取走最后一块棋 子 的棋手获胜。

问题 E. 扑克牌

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 兆字节

著名的间谍和特工詹姆斯-邦德终于成功找到了他现在的克星--兹洛博士。他们在 Permutasino 赌场的轮盘赌桌上相遇，现在 007 试图猜出他的恶敌的最新邪恶计划。

轮盘赌桌由 n 个单元组成，包含 1 到 n 之间整数的所有可能排列。每个赌注都是一个非负数 b_π，其中 π 表示它所对应的排列。所有投注的总和应等于 1。

下注后，轮盘机制会从数字 b_π定义的分布中随机选择一个排列。从形式上看，如果有赌注押在排列π上，则排列π的选择概率为 b_π，否则概率为 0。

为了在这场思想斗争中击败兹洛博士，说服他说出自己的邪恶计划，007 必须以这样的方式下注，即所产生的排列组合的预期值是一个向量 $(x_1, x_2, \ldots, x_n)$。将长度为 n 的随机排列的期望值定义为长度为 n 的向量，其中第 i 个元素是随机排列中*第* i 个元素的期望值。

帮助詹姆斯-邦德下适当的赌注，或者确定这是不可能的，这次他必须用不同的方法（比如射杀所有人并炸毁途中的每一栋建筑）来拯救世界。

输入

第一行输入包含一个整数 n，即轮盘赌桌上出现的排列长度（$1 \le n \le 500$）。

第二行包含 n 个整数 $x_1, x_2, \ldots, x_n$（$1 \le x_i \le n$），即所得排列的期望值。

输出

如果无法获得向量 $(x_1, x_2, \ldots, x_n)$ 作为博弈结果的期望值，则打印 -1。否则，在第一行打印 k（$1 \le k \le n$），即詹姆斯-邦德下注的次数。

在下面 k 行中的*第* i 行，打印赌注值 b_{π_i} ($0 \le b_{\pi_i} \le 1$) 和 n 个整数 $\pi_{i,1}, \pi_{i,2}, \ldots, \pi_{i,n}$ ($1 \le \pi_{i,j} \le n$，所有 $1 \le j \le n$ 上的所有 $\pi_{i,j}$都是不同的)，定义相应的置换 π_i.

总和 $b_{\pi_1} + b_{\pi_2} + \ldots + b_{\pi_k}$ 应等于 1，绝对误差不超过 10^{-6}。最大 $| b_{\pi_1} \pi_{1,j} + b_{\pi_2} \pi_{2,j} + \ldots + b_{\pi_k} \pi_{k,j} - x_j |$ 在所有 $1 \le j \le n$ *中的*值不应超过 10^{-2}。所有计算验证将使用双精度浮点数据类型进行。

如果有多个可能的答案，请打印其中任意一个。

示例

标准输入	标准输出
4 2 2 3 3	3 0.5000000000 1 2 3 4 0.1666666667 1 4 3 2 0.3333333333 4 1 3 2
2 1 1	-1

注

在第一次抽样检验中，所得排列的期望值为

$$\left(\frac{1}{2}\cdot 1+\frac{1}{6}\cdot 1+\frac{1}{3}\cdot 4,\ \frac{1}{2}\cdot 2+\frac{1}{6}\cdot 4+\frac{1}{3}\cdot 1,\ \frac{1}{2}\cdot 3+\frac{1}{6}\cdot 3+\frac{1}{3}\cdot 3,\ \frac{1}{2}\cdot 4+\frac{1}{6}\cdot 2+\frac{1}{3}\cdot 2\right) = (2, 2, 3, 3)。$$

问题 F. 平面最大切割

输入文件：	*标准输入*
输出文件	*标准输出*
时间限制	6 秒 内存限制
	512 兆字节

有些人可能知道如何在图中找到最小切点（Minimum Cut）。有些人可能还听说过 "最大剪切 "问题是 NP-完全的（更正式地说，寻找值至少为 k 的剪切的决策版本是 NP-完全的）。我敢打赌，你们中的一些人甚至想过这样的问题：嗯，如果我简单地否定所有代价，把 "最大切割 "问题转化为 "最小切割 "问题呢？我的图灵奖和十亿美元呢？哦，对了，我想到了"。

事实证明，对于某些有限类别的图形，问题可能比一般情况下更容易解决。平面最大切割问题的表述如下。考虑一个无向图 G，它由 n 个顶点 $V=\{v_1, v_2, \ldots, v_n\}$ 和 m 条边 $E=\{(v_{(a)(1)}, v_{(b)(1)}), (v_{(a)(2)}, v_{(b)(2)}), \ldots, (v_{(a)(m)}, v_{(b)(m)})\}$。这个图是平面图，我们给出了它在平面中的嵌入：除了图的描述，我们还知道顶点的坐标，这样，如果我们画出与图的边相对应的线段，没有两条边会共享一个内部点。此外，还有一个整数成本 c_j 与图中的 j 条边相关联。

你的任务是将图中的顶点划分为两个不相交的集合 A 和 B（$A\cup B=V$），使得切割边的总成本尽可能最大。如果一条边的一个端点属于 A，另一个端点属于 B，那么这条边就是切边。

输入

第一行输入包含两个整数 n 和 m（$1\leq n\leq 200$，$1\leq m\leq 1000$），分别是图的顶点数和边的数量。

下面 n 行中的第 i 行包含两个整数 x_i 和 y_i($-10^4\leq x_i, y_{(i)}\leq 10^4$)，即平面嵌入中第 i 个顶点的坐标。没有两点重合。

下面 m 行中的第 j 行包含三个整数 a_j、b_j 和 c_j（$1\leq a_{(j)}, b_j\leq n, a_j\neq b_{(j)}, 0\leq c_j\leq 10^5$）、第 j 条边的端点和相关费用。

输出

在第一行打印切割边的最大可能总成本。

在第二行，打印 n 个整数 $s_1, s_2, \ldots, s_{(n)}(s_i\in\{0, 1\})$ 其中，如果 $i\in A$，则 $s_i=0$；如果 $i\in B$，则 $s_i=1$。如果有多个可能的答案，则打印其中任意一个。

示例

标准输入	标准输出
4 5 0 0 2 0 0 2 2 2 1 2 3 2 4 6 3 4 4 1 3 7 2 3 8	21 0 0 1 1

问题 G. 大逃杀

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 兆字节

你玩过大逃杀游戏吗？在这类游戏中，玩家被单独空置在一个大区域内，他们必须收集武器、弹药、医疗包和盔甲，以杀死对方。当只剩下一名玩家时，游戏结束，这名玩家即为获胜者。

为了激起玩家之间的战斗，引入了安全区。在安全区之外的玩家会受到伤害。安全区总是越来越小，直到变成一个点，因此每个玩家迟早都要做出选择：进入安全区与其他玩家战斗，还是留在安全区外，损失命中点数并最终死亡。

在一维世界中，有一款著名的大逃杀游戏：未知战场》（BattleUnknown's Playergrounds）。游戏区域是一段 $[L, R]$。最初，安全区是整个$[L, R]$段，到游戏结束时，安全区将缩小到一个点 M（$L \le M \le R$），其中 M 在$[L, R]$段上等距离选择。安全区缩小的方式是 $\frac{M-L}{R-M} = const$，每秒钟长度减少 1。

今天，我们遇到了一场 n 名极其被动的玩家之间的比赛：其中第 i 名玩家躲在坐标为 x_i的房子里，而且不会移动。试想一下，他们宁可死在安全区之外，也不会离开自己的房子！他们的计划是利用收集到的药箱，让所有其他人都活下来：第 i 个玩家可以在安全区外活 a_i 秒。

请计算每位玩家赢得游戏的概率。

输入

第一行包含三个整数 n、L 和 R：玩家人数和游戏段落的终点（$1 \le n \le 10^5, -10^6 \le L < R \le 10^{(6)}$）。

下一行包含 n 个整数 x_i：玩家藏身房屋的坐标（$L < x_i < R$）。它们都是不同的，并按升序排序。

下一行包含 n 个整数 a_i：第 i 位玩家在安全区外的存活时间（$0 \le a_i \le 10^6$）。

输出

输出 n 行。第 i 行应包含一个实数：第 i 位玩家赢得游戏的概率。每个数字的绝对误差不应超过 10^{-9}。

示例

标准输入	标准输出
2 -5 5 -1 1 3 5	0.438447187191170 0.561552812808830
2 0 10 5 7 5 2	1.000000000000000 0.000000000000000
3 0 10 3 4 7 3 4 8	0.109584240176570 0.121686455414586 0.768729304408844

问题 H. 危险

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 兆字节

你们正在进行智力游戏的最后一轮。一共选择了 n 个题目，n 位作者每人为每个题目准备了一个问题。对于每个题目 i 和每个作者 j，你都知道自己回答第 j 位作者关于第 i 个题目的问题的概率。

在你真正得到问题之前，需要进行 $n-1$ 轮。在每一轮中会有两个动作。首先，你会丢弃任何一个话题的所有问题。其次，主持人会丢弃来自任何一位作者的所有问题。最终只剩下一个问题。如果你答对了，你就赢了这一轮。

当然，您弃列的目的是最大限度地提高您获胜的概率。相反，主持人则试图将概率降到最低。

如果你们都以最佳方式下棋，那么你们回答剩余问题的概率是多少?

输入

第一行是一个整数 n（$1 \le n \le 500$），即题目和作者的数量。然后是 n 行，每行 n 个整数。第 i 行对应第 i 个主题，上面的第 j 个数字是您将由第 j 位作者回答问题的概率。

概率以百分比表示。题目描述中的所有整数都介于 0 和 100 之间。

输出

打印一个数字：所需概率的百分比。

示例

标准输入	标准输出
2 1 100 99 0	1
3 0 50 100 100 0 50 50 100 0	0

问题 I. 拖鞋

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 兆字节

俄罗斯谚语说："谁先醒来，谁就能得到拖鞋"。然而，在我们的校园里，这可没那么容易。你不仅要早到（否则所有拖鞋都会被抢走），还要遵守鞋柜的严格规定。

鞋柜看起来像一个 $n\times m$ *的*网格。每个单元格都包含一只拖鞋，或左或右。最初，每只拖鞋都会朝左、右、前或后四个方向中的一个方向旋转。可以任意选取相邻的一对拖鞋（不一定完全相同），将其中一只顺时针旋转 90°，另一只逆时针旋转 90°。当你发现一双拖鞋处于自然位置时，你就可以穿上它，最后高兴而温暖地离开。

一双拖鞋的*位置*是*自然的*，如果

- 它们的细胞共用一条边；
- 它们朝向同一个方向；
- 如果你沿着这个方向看这两个单元格，一个单元格在右边，另一个单元格在左边。另外，右边的格子里有一只右拖鞋，左边的格 子 里有一只左拖鞋。

非正式地讲，一个正常人可以很自然地跳进这双拖鞋。有关例子，请参阅 "注释 "部分。

假设每个人都是利他主义者，并以最佳方式执行，求最多有多少人可以穿上这双拖鞋离开。

输入

第一行输入的是两个整数 n 和 m（$1\le n$，$m\le 100$），即网格的尺寸。接下来的 n 行中，每一行都包含 m 个空格分隔的字符串，描述相应单元格中的拖鞋。第一个字符为 "L "或 "R"，分别表示左侧和右侧滑块。第二个字符是 "< "、"> "、"^ "或 "v "中的一个。这表示拖鞋最初分别朝左、朝右、朝前或朝后。

输出

打印一个整数：可形成的最大拖鞋对数。

示例

标准输入	标准输出
2 2 R^ L> L< R^	2
3 2 L^ R^ R< L< L< R>	2

注释

请看第一个样本。首先，我们旋转两只左拖鞋：上部逆时针，下部顺时针。其次，我们旋转两只上面的拖鞋：左逆时针，右顺时针。之后，我们就有了两双处于自然位置的拖鞋：两双在上排，两双在下排。

```
R^ L>        R< L>        Rv Lv
L< R^        L^ R^        L^ R^
```

下面是本例中另一种可能的操作序列，它将导致不同的最终结果。

```
R^ L>      R< Lv      R< L>
L< R^      L< R^       L< R
```

问题 J. 好、坏、丑

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 兆字节

这个问题本来应该有一个很长的关于狂野西部的传说，但作者没有及 时 写出来，所以请发挥你的想象力！

考虑一条数字线。=每轮开始时，你可以说 "+ "或"-"。之后，棋手会根据你说的话改变位置。更确切地说，如果你说 t，而玩家站在位置 x，那么他就会移动到位置 $x' = x + d_t$，其中 d_+ 和 d_- 是两个整数常数。

您不知道 p、d_0和 d_1的确切值，但您知道玩家是 "好人"、"坏人 "或 "丑人"（是的，想象力！）：

- 好玩家的 $p = m$，$d_+ = 2$，$d_- = -1$；
- 坏玩家有 $p = -m$，$d_+ = 1$，$d_- = -2$；
- 丑棋手有 $p = m$ 或 $p = -m$ 以及 $d_+ = 1$ 和 $d_- = -1$ 或 $d_+ = -1$ 和 $d_- = 1$。

如您所见，棋手的起始位置取决于某个整数常数 m（$1 \le m \le 1000$）......遗憾的是，您也不知道。

每轮比赛结束后，棋手会告诉你他现在的位置是否为 $x = 0$。

看来，只要玩上几轮，就能独一无二地判断出选手是好、坏还是丑。在不超过 30m 的回合内完成。

在每次测试中，值 m、p、d_+ 和 d_- 都是根据上述规则选择的。它们事先是固定的，在检查过程中不会改变。

交互协议

这是一个互动问题。

如果要进行一轮游戏，请在另一行打印"+ "或"-"。如果棋手到达了位置 $x = 0$，则会得到一行包含 1 的对话框；如果棋手站在其他位置，则会得到一行包含 0 的对话框。

如果您已准备好猜测玩家的类型，请打印一行包含字符"！"、空格和 "好"、"坏 "或 "丑 "中的一个单词。之 后 ，程序必须终止。

如果您在游戏 *30* 米后仍未提供答案，您的解决方案将得到 "错误答案 "的结果。

要防止输出缓冲，请在每打印一行后清空输出缓冲： ，例如，可以使用 C 或 C++ 中的 `fflush (stdout)`、Java 中的 `System.out.flush()`、Pascal 中的 `flush (output)` 或 Python 中的 `sys.stdout.flush ()` 来实现。

示例

标准输入	标准输出
 0 1	- - 好