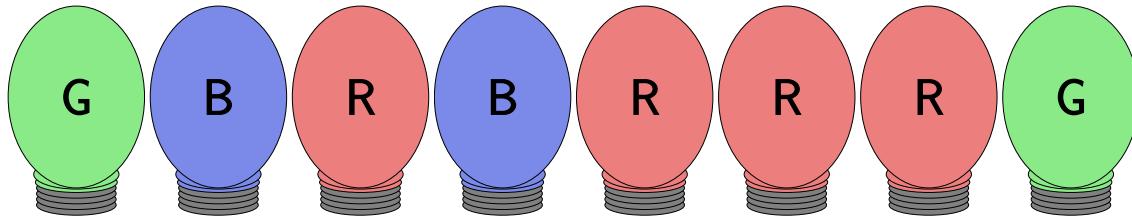


问题P

转为红色时间限制：3秒

梅的父母在过去的一年里对他们的房子进行了翻修，但是他们的照明系统相当复杂！房子里的每个房间都有一个LED灯，可以设置成红色、绿色或蓝色，如图P. 1所示。



图P. 1：示例输入1中灯的初始状态。按钮和导线未显示。

整栋房子里布满了各种按钮，每个按钮都连接着一个或多个灯。按下按钮后，与该按钮相连的红色灯变为绿色，绿色灯变为蓝色，蓝色灯则变为红色。每个按钮可以多次按下。由于这栋房子是在横梁布线技术发明之前建造的，因此每个灯最多只能由两个按钮控制。

梅最喜欢的颜色是红色，所以她想把所有的灯都变成红色。她的父母担心按钮会磨损，要求她尽量减少按按钮的次数。

输入

输入的第一行包含两个正整数l 和 b，其中l (1 ≤ l ≤ 10^5) 表示灯的数量，b (0 ≤ b ≤ l) 表示按钮的数量。第二行是一个由l 个字符组成的字符串，这些字符全部是R、G或B，第i 个字符表示第i 个灯的初始颜色。接下来的b行描述了按钮。每一行以一个整数k (1 ≤ k ≤ l) 开头，表示该按钮控制的灯的数量。然后是k个不同的整数，表示该按钮控制的灯。灯从1到l 编号，包括两端。每个灯在所有按钮中最多出现两次。

输出

输出Mei 需要按下按钮的最小次数才能将所有灯都变成红色。如果Mei 无法将所有灯都变成红色，则输出“不可能”。

样本输入1

```
8 6 GBRBRRRG2 1 4
1 2
4 4 5 6 7
3 5 6 7
1 8
1 8
```

示例输出1

```
8
```

样本输入2

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

示例输出2

不可能的

样本输入3

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

示例输出3

6

样本输入4

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

示例输出4

3

问题0

进行容器洗牌时间限制：2秒

威风凛凛的货船，每艘都载着成千上万的船只集装箱，每天在世界的海洋上航行。它们创造了现代通过如此高效地运输货物到一半的距离，可以实现贸易在世界各地，这只需要几美分。

一旦船只抵达目的地，它们的标准尺寸货柜就会从船上卸下，堆放在陆地上，然后被转移到火车或卡车上，最终运送到目的地。事实证明，搬运集装箱非常昂贵，因此港口运营商会尽量减少货物运输所需的移动次数。



马蒂尼171通过维基共享资源，署名-相同方式共享的货物集装箱

在这个问题中，我们考虑这样的容器卸载场景。我们需要卸载n个集装箱，这些集装箱被放置成从下到上的两层堆栈。每个集装箱的放置是随机的，有相等的概率会被放在第一层或第二层（与其他集装箱无关）。一旦所有集装箱都卸载完毕，它们将按照给定的顺序由卡车装载。当一辆卡车想要装载特定的集装箱时，有两种情况。如果集装箱位于堆栈的顶部，则可以将集装箱移至卡车，而无需移动任何其他集装箱。否则，必须将集装箱从一个堆栈移动到另一个堆栈，直到所需集装箱位于其堆栈的顶部。此时，可以将集装箱移至卡车上。

例如，考虑三个容器按顺序1、2、3到达的情况。假设1和3在第一堆中，2在第二堆中。如果容器按照1、2、3的顺序被搬到卡车上，则需要进行五次容器移动：

第1层	第2层	评论
1 3	2	初始配置（从下到上）
1	2 3	将容器3从堆栈1移动到堆栈2
	2 3	将容器1移动到卡车
3	2	将容器3从堆栈2移动到堆栈1
3		将容器2移动到卡车
		将容器3移至卡车

表0.1：订单1、2、3中要求的集装箱移动示例。

我们想知道将所有集装箱运送到客户手中需要多少次移动。假设集装箱的放置是随机的，我们要求您计算给定卡车装载订单所需的预期移动次数。

输入

第一行输入包含一个整数n ($1 \leq n \leq 10^6$)，表示集装箱的数量。集装箱编号为1；2；...；n，按照此顺序从船上卸载。第二行输入包含n个整数a1；...；an。这些数字是集合{1；2；...；n}的排列，指定了集装箱装载到卡车上的顺序。

输出

输出将集装箱装载到卡车上的预期移动次数，这不包括从船上卸载集装箱的成本，但包括堆垛之间的移动和从堆垛到卡车的移动。您的答案的绝对误差应不超过 10^{-3} 。

样本输入1

5 4 2 5 3 1	7.000
----------------	-------

示例输出1

样本输入2

6 1 2 3 4 5 6	13.500
------------------	--------

示例输出2

问题R动物园管理时间

限制：5秒

管理动物园时，有时需要将动物在不同的围栏之间移动。你可能会发现斑马更喜欢企鹅目前所在的较宽敞的围栏，而企鹅可能想搬到考拉现在居住的较冷的围栏；考拉则会搬到一个空闲的围栏，那里可以种植桉树。因此，你会先将考拉移走，腾出较冷的围栏，然后将企鹅移到那里，最后再将斑马移走。

你移动动物的方式是使用连接围栏的特殊隧道——你不希望动物移动到外面，一方面是因为它们可能会被吓到，另一方面是因为它们可能会逃跑并伤害自己。

不幸的是，您最近获得了更多的动物，所有的围栏现在都满了，这使得移动动物变得更加困难。例如，想象一下，如果考拉要搬到以前的斑马围栏——您不能先移动任何一组动物。相反，你学会的是同时移动动物——斑马、考拉和企鹅同时开始沿着不同的隧道移动，并且同时到达新的围栏——因此它们永远不会相遇。请注意，你不能仅仅通过这种方式交换两个相连围栏中的动物（因为它们会在隧道中相遇并感到害怕）。

所以，现在你有一个谜题。你有一系列的围栏，每个围栏里都关着一种动物；有些围栏之间通过隧道相连。你可以多次选择一组动物，并将每只动物移动到相邻的围栏中，只要该围栏里的动物也在同一移动中被移动，并且同一个移动中不能重复使用一条隧道。你还有一套关于如何完美地安排这些动物的方案。是否可以通过一系列的移动来实现这一点？

输入

输入的第一行由两个整数 n ($1 \leq n \leq 105$) 和 m ($0 \leq m \leq 105$) 组成，表示围栏和隧道的数量。然后是 n 行，其中第 i 行包含两个整数 b_i ($1 \leq b_i \leq 10^6$) 和 e_i ($1 \leq e_i \leq 10^6$)，表示初始时围栏 i 中动物的种类，以及你希望所有移动后围栏 i 中的动物种类。你可以假设 $1, \dots, e_n$ 是 b_1, \dots, b_n 的排列。

然后是描述隧道的 m 行。每一行包含两个整数 x 和 y ($1 \leq x < y \leq n$)，表示围栏 x 和围栏 y 通过一条双向隧道相连。没有两个围栏通过多于一条隧道相连。

输出

如果可以移动动物，使每个围栏都包含所需的动物类型，则输出为可能。否则，输出为不可能。



一张动物园的照片，里面有骆驼、驴和（部分）山羊。

样本输入1

```
3 3
1 4
4 7
7 1
1 2
2 3
1 3
```

示例输出1

做得到的

样本输入2

```
2 1
1 2
2 1
1 2
```

示例输出2

不可能的

样本输入3

```
5 6
10 40
20 30
30 50
40 20
50 10
1 2
2 3
1 3
3 4
3 5
4 5
```

示例输出3

做得到的

样本输入4

```
4 4
10 10
10 20
20 10
20 20
1 2
2 3
3 4
1 4
```

示例输出4

不可能的

问题S桥接差距时

限：4秒

一群步行者在夜里来到河边。他们想过一座桥，这座桥只能容纳有限的人数每次只有一名步行者。步行者只有一支火炬，过桥时需要使用的东西。每个步行需要一定的时间才能通过；一群人通过大家必须以最慢的人的速度一起走。什么是所有步行者过马路的最短时间

桥

例如，示例输入1假定桥可以一次可以容纳2名步行者，共有4名步行者穿越时间1分钟、2分钟、5分钟和10分别用时17分钟

可以通过以下交叉顺序实现。

首先，两个最快的步行者在2分钟内相遇。其次，最快的步行者在1分钟内返回。第三，两个最慢的步行者在10分钟内相遇。第四，第二快的步行者在2分钟内返回。第五，两个最快的步行者在2分钟内相遇。



低容量桥梁

输入

第一行输入包含两个整数n和c，其中n (2 ≤ n ≤ 104) 是行人的数量，c (2 ≤ c ≤ 104) 是桥一次能容纳的行人数量。

接着是一行包含n个整数t1, ..., tn (对于所有i, 1 ≤ ti ≤ 109) 的数字。第i个行走者需要时间ti来穿过。

输出

输出整个团队过桥所需的最短总时间。

样本输入1

示例输出1

4 2	17
1 2 10 5	

样本输入2

示例输出2

4 6	10
1 2 10 5	

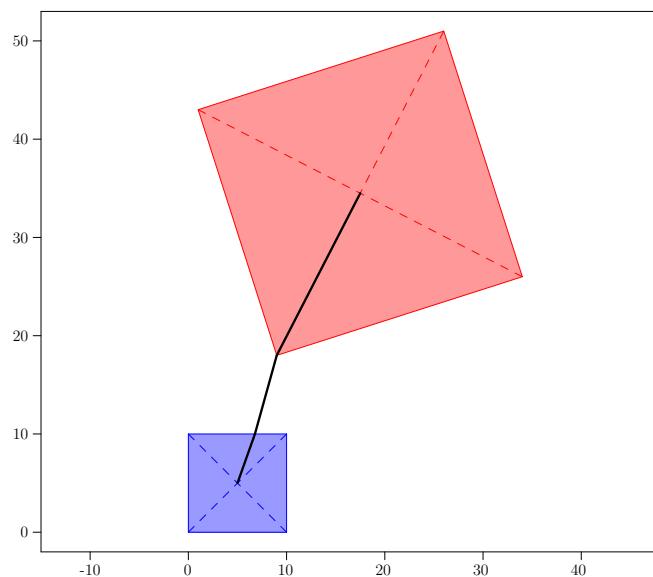
本页有意留白。

问题T卡尔的假期

时限 : 1秒

卡尔蚂蚁回来了！在蜿蜒的小径上穿行（2004年世界总决赛问题A）和在八面体上漫步（2009年世界总决赛问题C）之后，是时候小憩一下了——是时候去看看风景了！而还有什么地方比高耸的金字塔尖更适合看风景呢！！而还有什么地方比埃及更适合看高大的金字塔呢！！！（这真是太令人兴奋了！！！！！）

欣赏完一座金字塔顶端的景色后，卡尔想前往另一座金字塔的顶端。由于蚂蚁在烈日下表现不佳，他希望找到这两座金字塔顶端之间最短的距离，假设他只能在金字塔表面和它们所在的平面上行走。这两座金字塔在几何上是直角方形金字塔，即金字塔的顶点正好位于方形底面中心的正上方。



图T.1：对应于样本输入1的两个金字塔的示意图。黑色线表示它们顶点之间的最短路径。

输入

第一行输入包含五个整数 $x_1; y_1; x_2; y_2; h$ ，其中 $x_1; y_1; x_2; y_2$ (-105 $\leq x_1; x_2; y_1; y_2 \leq 105$) 和 $(x_1; y_1) \neq (x_2; y_2)$ 定义了第一个金字塔的一条边，金字塔的主体位于从 $(x_1; y_1)$ 到 $(x_2; y_2)$ 的有向量左侧， $h (1 \leq h \leq 105)$ 是金字塔的高度。第二行输入以相同格式描述第二个金字塔。两个金字塔底面的交点面积为0。

输出

输出卡尔在两座金字塔尖端之间行走的最小距离。你的答案的绝对或相对误差应不超过 10^{-6} 。

样本输入1

0 0 10 0 4	示例输出1
9 18 34 26 42	60.866649532

示例输出1

0 0 10 0 4	示例输出1
9 18 34 26 42	60.866649532

问题U玩具火车轨道

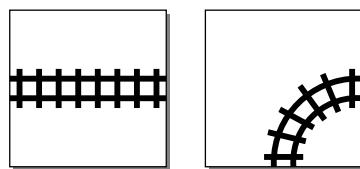
时间限制：1秒

每个小孩，甚至不少大人，都对玩具火车着迷。从幼儿的小小火车到爱好者精心制作的模型铁路，填满整个地下室，它们都是有利可图的生意。玩具火车轨道建设公司（TTTCC）为所有年龄段和技能水平的用户制造火车轨道。为了保持现有客户的活跃度并吸引一些新客户，TTTCC最近开始发布如何将轨道连接成复杂布局的地图。通常，这始于设计师构思出一个有趣的轨道布局，然后发布该布局以及所需的不同轨道段（如弯道和直线部分）的数量。


模型铁路，由Marshman通过Wikimedia Commons提供，采用CC-by-sa许可。

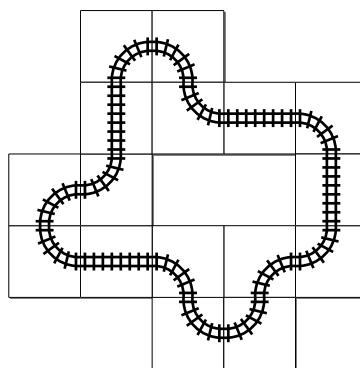
但是，TTTCC最近了解到许多客户正在寻找相反的情况：他们已经有一些轨道段（可能是从奶奶的阁楼里找到的），并希望用它们来创建一个大型的火车线路。这有多难呢？

为了研究自动化布局创建过程的可行性，TTTCC对使用两种不同的形状来构建列车路线很感兴趣：直线段和90度转弯（见图U. 1）。



图U. 1：直线轨道段和曲线轨道段。

有效的布局是通过将这些形状放置在一个正方形网格上创建的，每个轨道部件恰好占据一个网格单元。两种类型的部件都可以以90度增量旋转。“正确”的火车轨道需要连接起来，并形成一个单一的闭合环。给定一组直线和曲线轨道段，可以构建的最长闭合环是什么？



图U. 2：使用四个直线段和十二个曲线的示例轨迹。这对应于示例输出1。

输入

输入由包含两个整数s和c的单行组成，分别表示可用直线和曲线轨道段的数量（0 ≤ s ≤ 105, 0 ≤ c ≤ 105）。

输出

输出一个最多由s条直线段和c条曲线段组成的火车环路，该环路在限制条件下具有最长的长度（以使用的轨道段数计算）。环路必须闭合且不能自相交。如果有多个最大长度的环路，任何一个都将被接受。

如果环的长度为n，则打印一个长度为n的字符串，其中字符表示单次遍历中遇到的环段。字符‘S’表示直线段，‘L’表示左转的曲线段，‘R’表示右转的曲线段。

样本输入1

4 12	LSRLLRLSLSRLLSRL
------	------------------

示例输出1

样本输入2

1 5	LLLL
-----	------

示例输出2

问题三：三种骰子

时限：1秒

看看他们怎么掷骰子！根据一个著名的故事，沃伦·巴菲特曾向比尔·盖茨发起了一场简单的骰子游戏挑战。他有三个骰子；第一个玩家可以检查并选择其中一个。第二个玩家则从剩下的两个中选择一个，然后双方将骰子相互投掷，目标是获得最高的数字。沃伦提出让比尔先掷，但这让比尔起了疑心，所以他选择了后掷。事实证明这是一个明智的选择：这些骰子是不可传递的。当第一个骰子与第二个骰子对掷时，第一个骰子占优势；当第二个骰子与第三个骰子对掷时，第二个骰子占优势，但当第一个骰子与第三个骰子对掷时，第一个骰子并没有优势！



来自rawpixel 的图像，CC0

为了正式化这一点：定义“骰子”为任何至少有一个面显示正整数的形状。当掷骰子时，其一个面会被均匀随机选择。当两个骰子相互投掷时，被选中面显示较大数字的骰子得1分；如果两个数字相等，each die earns $\frac{1}{2}$ points. 对于骰子D和D'，定义得分(D, D')为单次掷骰子时D获得的预期点数。如果得分(D, D') > $\frac{1}{2}$, we say that D has an advantage over D'; if 评分(D, D') = $\frac{1}{2}$, the two dice are tied. For example, if D is the first die in the sample input and D' is

第二，分数(D, D') = $\frac{4}{9}$ and score(D', D) = $\frac{5}{9}$ ，所以D比D'有优势。

给定两个骰子D1和D2，其中D1比D2有优势，你希望有一个第三颗骰子D3，与前两颗骰子形成一个不可传递的组合。在所有比D1有优势或与D1平手的D3中，计算出最低可能的得分(D3, D2)。If this is less than $\frac{1}{2}$, you can make an intransitive三重奏！同样地，在所有D3中，如果D2对D3有优势或与D3相同，则计算最高可能得分(D3, D1)。

输入

输入包含两行，每行描述一个骰子。其中一个骰子（第一个或第二个）比另一个有优势。具有优势的骰子是D1，另一个是D2。

线上的第一个整数给出n(1 ≤ n ≤ 105)，骰子的面数。然后是n个整数fi(1 ≤ i ≤ n, 1 ≤ fi ≤ 109)，表示每个面上的数字。

输出

输出一条线，包含上述条件下的最低分(D3, D2)和最高分(D3, D1)。两个分数不需要使用相同的骰子D3。您的答案的绝对误差应不超过10-6。

样本输入1

6 1 1 6 6 8 8
3 2 4 9

示例输出1

0.291666667 0.750000000

样本输入2

4 9 3 7 5	示例输出2
3 4 2 3	0.500000000 0.500000000

问题W斯芬克斯之谜时间

限制：2秒

埃及最引人注目的地标之一是吉萨的大斯芬克斯，这是一座描绘了神话生物的雕像，它有着人类的头、狮子的身体和鹰的翅膀。在埃及和希腊神话中，斯芬克斯被视为守护者。最著名的斯芬克斯可能是守护希腊城市底比斯的那个。据传说，当俄狄浦斯试图进入这座城市时，斯芬克斯给他出了一个谜语：哪种生物早晨有四条腿，下午有两条腿，晚上有三条腿？正如你可能听说过的那样，俄狄浦斯正确地回答了：人——婴儿时四肢爬行，成年后用两条腿行走，老年时则使用拐杖。

在这个问题中，你遇到了一个不同的斯芬克斯，他给你出了一个有点反向的谜题：阿克希、巴塞利斯克和半人马各有多少条腿？虽然你知道这些是来自埃及和希腊神话的生物，但你完全不知道它们各自有多少条腿（只知道是非负整数）。狮身人面像严厉地指示你不要碰任何东西，所以你无法在手机上搜索答案。

然而，斯芬克斯允许你向她提问。在每个问题中，你可以问斯芬克斯这些生物总共有多少条腿。例如，你可以问，三只巴塞利斯克和一只阿克西斯一共有多少条腿？或者五匹半人马一共有多少条腿？这看起来很简单，但你很快意识到斯芬克斯是狡猾的生物：斯芬克斯的回答中可能有一个完全的谎言，而你无法分辨哪个是假的。



古斯塔夫·莫罗的《俄狄浦斯与斯芬克斯》，1864年，公共领域

编写一个程序与斯芬克斯对话，询问ve问题，并解决谜题。

互动

共有五轮问题。每轮问题中，你首先需要写一行包含三个用空格分隔的整数a、b和c（0 a, b, c 10），表示问题是“a只阿克塞、b只巴塞利斯克和c只森陶尔总共有多少条腿？”问题提出后，标准输入会提供一行包含一个整数r（0 r 10^5）的输入行，给出斯芬克斯对你的问题的回答。

在ve轮询问之后，输出一行，包含三个空格分隔的非负整数la、lb和lc，分别表示轴、蛇怪和半人马的腿数。

阅读

样本交互1

写入

12

1 1 1

13

1 1 1

24

5 0 1

4

1 0 0

8

1 1 0

4 4 4

阅读

样本交互2

写入

2023

4 4 4

0

1 0 0

42

0 1 0

2024

0 0 1

0

0 0 0

0 42 2024

问题X四重奏

时限：2秒

你正在观看一群孩子玩四人组游戏，玩得很开心。突然间，你开始怀疑有些孩子可能在作弊。孩子们似乎对此毫不在意。相反，这反而让他们更加开心，特别是当有人被发现作弊时。作为一名程序员，你立刻开始思考如何仅通过观察他们的游戏来检测作弊行为。

四重奏是一款四人纸牌游戏，使用一副32张的牌，分为8组，每组4张。实际的卡片通常包含教育图片，这有助于玩家不仅学习单个物体，还能了解它们的分类。例如，卡片可能展示动物，而这些组则对应不同的动物类别（哺乳动物、爬行动物等）。

在四重奏游戏开始时，每位玩家会获得8张牌，玩家1开始他们的回合。每回合，一位玩家会询问另一位玩家是否拥有一张特定的牌。如果被点名的玩家持有该卡牌，他们必须将其交给请求者，然后请求者通过向其他玩家索要另一张卡牌来继续自己的回合。如果被点名的玩家没有持有的请求卡牌，请求者将失去本轮，而被点名的玩家开始自己的回合。一个重要的规则是，请求者必须已经拥有一张与他们所持卡牌同组的卡牌。



一套四张牌，来自威廉·布什的德国四重奏牌组，通过维基共享资源，公共领域

请求，寻求

如果轮到某位玩家时手中持有完整的四张牌（“四张牌”），该玩家可以向其他玩家展示这四张牌，将其放在一边并获得一分。四张牌将永久从当前游戏中移除。如果玩家任何时候没有剩余的牌，则该玩家退出游戏。如果轮到这位玩家，接下来的玩家按顺序（1-2-3-4-1-...）中仍持有牌的玩家将接替其位置。如果没有玩家持有牌，游戏结束，得分最高的玩家被宣布为胜者。

玩家不允许要求从游戏中移除的卡牌，他们也不能要求离开游戏的玩家的卡牌。

在四人组游戏中作弊可以想象为两种情况，即玩家虚假声称自己拥有（或不拥有）某张牌。具体来说，如果

- ? 玩家要求一张牌，尽管他们没有相应套牌的牌，或者
- ? 被要求的玩家声称没有要求的卡，尽管他们有。

不用说，作弊通常是在游戏结束之后才被发现的。理论上，一个警惕的对手最终会发现任何作弊行为，因为所有的牌在游戏结束前的某个时刻都会被揭开。

请注意，要求玩家不可能拥有的卡片（例如，因为请求者自己手中有这张卡片）并不是一个明智的举动，但本身并不被认为是作弊。

输入

第一行输入包含一个整数n (1 ≤ n ≤ 1000) , 表示从游戏开始时起 , 在一场四重奏游戏中连续进行的动作次数。接下来的n行每行描述一个动作。每个动作由一行描述 , 可能包括以下几种 :

- ? x A和sk yes——玩家x向玩家y要sk这张牌 , 玩家y把这张牌交给玩家x ;
- ? x A和sk—玩家x向玩家y要求卡片sk , 玩家y声称没有这张卡 , 并开始他们的回合 ;
- ? x 0 s-玩家x留下一个四重奏s。

在所有动作中 , x和y (1 ≤ x, y ≤ 4 , x ≠ y)是表示玩家的整数 , s (1 ≤ s ≤ 8)是一个一位数整数 , 对应于一组/四张牌之一 , 而k {A, B, C, D }是一个字符 , 用于标识一组中的四张牌。因此 , 这些牌被标记为1A、1B、1C、1D、2A、2B , … , 8C、8D。具有相同首位数字的四张牌构成一个单独的组。

给定的序列描述了4名玩家之间进行的一场有效游戏。也就是说 , 除了上述两种可能的作弊方式外 , 所有行动都符合所有规则。

输出

如果有初始分配的卡片 , 使得行动序列对应于一个 (可能是部分的) 游戏 , 所有玩家都遵守规则 , 则输出 “是”。否则 , 输出 “否” , 并附上一行显示第一个行动之后 , 可以确定某个玩家必定作弊的行动编号 , 行动按顺序编号 , 从1开始 , 放置四张牌的行动也计入其中。

示例输入1的说明

玩家1和玩家2都声称没有3C , 而且他们都没有3A或3D。所以其中一人要么是在谎称没有3C , 要么是要求3C但没有3B (第3套中唯一剩下的卡牌)。

样本输入1

```
4
1 A 2 3C否
2 A 3 A是
2 是的 , 4D
2 A 1 3C否
```

示例输出1

```
不
4
```

样本输入2

```
6
1 A 2 3C否
2 A 3 A是
2 是的 , 4D
2 A 1 3B否
1 A 4 5B是
1 Q 5
```

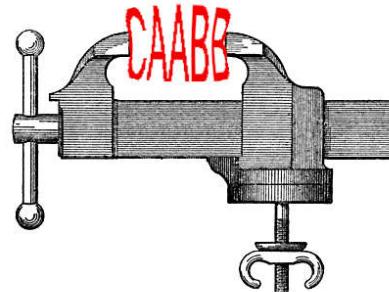
示例输出2

```
对
```

问题Y

压缩时间限制：1秒

无限压缩计划联盟 (ICPC) 开发了一种新的、出色的数据压缩策略，称为“不要重复自己” (DRY)。DRY作用于字符串，如果字符串中包含两个连续相同的子串，它会简单地移除其中一个。例如，在字符串“苜蓿种子”中，它可以移除“种子”中的两个“e”子串之一，以及“苜蓿”中的两个“lfa”子串之一，结果是“阿尔法塞德”。DRY还可以利用之前的移除——例如，在字符串“第十七次行李”中，它首先移除“第十七次”中的重复“e”和“行李”中的重复“g”，结果是“第十七次行李”，然后移除重复的



图片改编自维基共享资源，公共领域

将“seventeenth”中的“ent”和“baggage”中的“ag”折叠，得到“seventh bage”。

如果存在多个重复连续子字符串以删除的选择，DRY应选择能够产生最短最终字符串的方式。例如，在字符串“abbc当地abcccd”中，DRY有两种选择——要么去掉开头附近的重复字母“B”，要么去掉末尾的重复字母“CD”。如果先去掉“B”，那么DRY就可以去掉重复的“BCDC”，得到“ABCD”，再从这个“ABCD”中去掉末尾的“CD”，最终得到“ABCD”。然而，如果DRY先去掉末尾的“CD”，则会剩下“abbc当地bcd”，此时只能去掉重复的“B”才能得到“abbc当地bcd”——而这个字符串再也不能被进一步压缩了。因此，DRY的正确选择是先压缩双字母“B”，最终得到“ABCD”。

ICPC发现DRY在处理二进制字符串时效果最佳——即仅由零和一组成的字符串。因此，他要求你实现一个尽可能优秀的DRY算法来处理二进制字符串。对于任何二进制字符串，你应该输出通过反复应用DRY能够得到的最短字符串。

输入

输入由单行组成，包含长度小于或等于105的非空字符串，仅由零和一组成。

输出

输出一行，包含对输入字符串运行DRY的最短可能结果。如果有多个可能的最短结果，则接受其中任何一个。

样本输入1

1111	1
------	---

示例输出1

样本输入2

101	101
-----	-----

示例输出2

样本输入3

10110

示例输出3

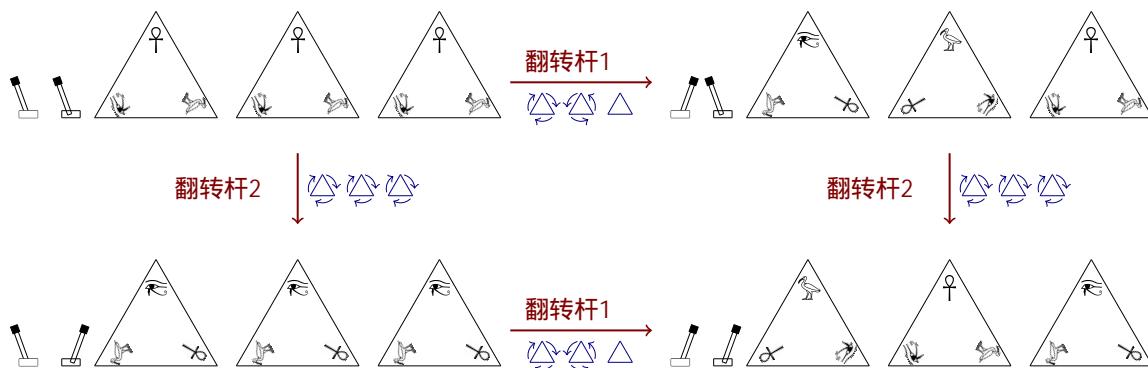
10

问题Z考古发掘

时限：1秒

埃及学著名教授Z·穆默正在探索卢克索新发现的一座陵墓，在那里她发现了一处神秘的建筑。墙上排列着一排金字塔形状的石板。每块石头上都有三个象形文字：一个安卡（顶部），一个荷鲁斯之眼（左下角），以及一只朱鹭（右下角）。

墙边有n个杠杆。教授小心翼翼地尝试这些杠杆，警惕潜在的陷阱，发现每个杠杆都能使一些金字塔顺时针或逆时针旋转，从而使另一个象形文字朝上。将杠杆翻转回去，则会使相同的金字塔反向旋转回原位。这些杠杆完全独立运作，因此无论其他杠杆处于什么状态，推动或撤回一个杠杆都会产生相同的效果。见图Z.1示意图。



图Z.1：一个墙的示意图，其中 $k = 3$ 个金字塔和 $n = 2$ 个杠杆。第一个杠杆使第一个金字塔顺时针旋转，第二个金字塔逆时针旋转，而第三个金字塔保持不变。第二个杠杆使所有三个金字塔顺时针旋转。这对应于示例输入1。

出于好奇，穆默教授记录了每个杠杆的单独效果。探险归来后，她回到大学，给一名学生布置了一项艰巨的任务，即找出所有可能的金字塔配置（其中一些可能是相同的），这些配置可以通过调整部分杠杆来实现，以便进一步研究。

经过许多个夜晚的仔细计算，学生终于完成了作业，开始收拾他的文件。然而，灾难突然降临：他不小心洒了一些墨水，彻底毁坏了教授的原始笔记，这些笔记中包含了每个杠杆单独作用的唯一记录。

唯一能逃脱穆默教授怒火的方法是根据可能的金字塔配置列表重建原始笔记。这无法完全明确地完成（例如，无法区分同一组杠杆的不同排列顺序）。但只要操纵杆仍然产生计算出的金字塔结构列表，这个错误就不太可能被注意到（至少在学生毕业之前不会）。

输入

第一行输入包含三个整数n、k和t，其中n (1 ≤ n ≤ 40) 和k (1 ≤ k ≤ 5) 分别是杠杆和金字塔的数量，t (1 ≤ t ≤ 3k) 表示不同金字塔配置的数量。接下来的t行描述了这三种不同的可能金字塔配置。每一行由一个长度为k的字符串组成，描述了该配置，以及一个整数f (1 ≤ f ≤ 2n)，表示导致此配置的不同杠杆设置数量。配置使用字母‘A’、‘E’和‘I’描述。第j个字符表示第j个金字塔上朝上的象形文字：‘A’表示安克，‘E’表示荷鲁斯之眼，‘I’表示朱鹭。

给出的t个配置是成对不同的，所有t条线上的f值之和等于2n，输入使得至少有一个杠杆列表产生给定的多组配置。

输出

输出n行，描述产生给定的金字塔配置多重集的一组可能的杠杆。每行描述一个杠杆，使用长度为k的字符串，由符号‘+’、‘-’或‘0’组成。在每个这样的字符串中，第j个字符描述该杠杆对第j个金字塔的作用：如果杠杆使金字塔顺时针移动，则为‘+’；如果杠杆使其逆时针移动，则为‘-’；如果杠杆不移动这个金字塔，则为‘0’。

如果有多个可能的解决方案，任何一种都将被接受。

样本输入1

2 3 4	+ - 0
EEE 1	+++
EIA 1	
IAE 1	
AAA 1	

示例输出1**样本输入2**

3 2 2	- 0
IA 4	0 0
AA 4	0 0

示例输出2