

## Problem A. Airport Construction

Input file:           standard input  
Output file:          standard output  
Time limit:           2 seconds  
Memory limit:        1024 megabytes

位于热带的 Piconesia 岛国以其秀丽的沙滩、繁茂的植被、可可与咖啡原料的产出以及全年宜人的天气而闻名海外。天堂般美丽的这里将纳入未来举办 ACM-ICPC 世界总决赛的地点考虑名单中（至少也会是执行委员会度假地点的选择之一）。但是还有一个小问题：这个岛真的很难到达。

就目前而言，最快的方式是从最近的机场出发，并借助渔船、油轮、皮划艇和潜艇这些工具，花费三天时间才能到达小岛。为了让大家更轻松地完成 ICPC 世界总决赛，也为了推动小岛的旅游业发展，Piconesia 正在计划修建它的第一个机场。

考虑到越长的起降跑道越有助于对大型飞机的支持，Piconesia 决定在其岛上修建一条最长的起降跑道带。不幸的是，他们无法确定这条跑道应该设置在哪里。也许你可以帮个忙？

对于本题，我们将 Piconesia 的边界抽象成一个多边形的模型。根据这个多边形，你需要计算出最长的跑道长度（即最长的线段）使得跑道可以完全在小岛上修建。图 1 对应于第一个输入样例。

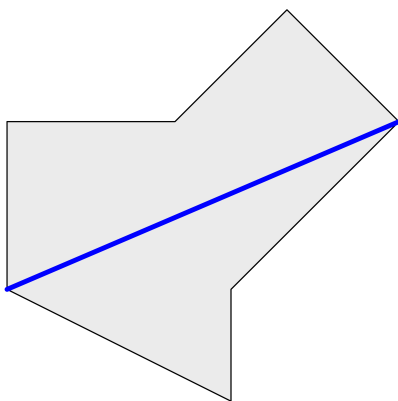


图 1. 小岛的多边形模型。最长的跑道已用粗线标出。

### Input

第一行包含一个整数  $n$  ( $3 \leq n \leq 200$ )，表示多边形的点数。

接下来  $n$  行，每行包含两个整数  $x$  和  $y$  ( $|x|, |y| \leq 10^6$ )，表示多边形上的顶点  $(x, y)$ ，顶点按照逆时针顺序给出。

给出的多边形是简单多边形，即它的顶点互不相同，并且除了相邻的两条边在公共点相交之外，它的任意两条边都不会相交。此外，相邻的两条边也不会共线。

### Output

输出能放进给定多边形内部的最长线段长度，其绝对误差或相对误差不能超过  $10^{-6}$ 。

**Examples**

standard input	standard output
7 0 20 40 0 40 20 70 50 50 70 30 50 0 50	76.157731059

standard input	standard output
3 0 2017 -2017 -2017 2017 0	4510.149110617

## Problem B. Get a Clue!

Input file:           standard input  
Output file:          standard output  
Time limit:           4 seconds  
Memory limit:        1024 megabytes

起源于 19 世纪 40 年代的英国的游戏 Cluedo 是世界上最流行的图版游戏之一。游戏的目标是找出谋杀 Boddy 先生的凶手，凶手使用的凶器，以及行凶的地点。游戏使用一系列卡片表示六个角色 (标号 A, B,  $\dots$ , F)、六种武器 (标号 G, H,  $\dots$ , L) 和九个房间 (标号 M, N,  $\dots$ , U)。在游戏的开始，角色、武器和房间分别会有一张卡片在玩家不知道的情况下被随机移除，它们分别表示凶手、凶器和作案地点。剩下的 18 张卡片会重新排列并分发给玩家，第一张会给玩家 1，接下来会给他右边的玩家 2，依次类推。一些玩家可能比其他玩家多拿一张牌。由于这道题的原因，玩家的数量为四个，这意味着玩家 4 的右边是玩家 1。

游戏剩下的部分便是寻找线索。玩家轮流决策，从玩家 1 开始，然后依次是其右边的玩家。每轮决策需要提出一个 *suggestion* (即提议，包括指定嫌疑人、凶器和作案地点) 并轮询其他玩家是否有证据能反对这个提议。例如，你可以对另一个玩家说：“我认为凶手是角色 A，凶器是 L，作案房间是 T。”如果其他玩家恰好有其中一张卡片，这个玩家必须向你 (且只有你) 展示这张卡片。如果他有多张这样的卡片，他可以任选一张向你展示。

当你提出提议时，你需要先问你右手边的玩家是否可以反驳你。如果他没有反驳，你可以继续问这个人右手边的玩家，直到有人提出反驳的证据，或是所有人都无法反驳你的提议。

许多时候你可以从其他人的提议过程中获取信息。例如上面的例子，你是拥有卡片 A 和 T 的第三方玩家。如果有人向提议者展示了反驳的证据，那么你可以知道那张卡片一定是武器 L。在游戏过程中，保持对提议与反驳过程的追踪是很重要的游戏策略。

为了赢得游戏，你需要进行 *accusation* (即指控)，做出你对凶手、凶器和作案地点的最终猜测。在你的指控开始后，你可以检查最开始被移除的三张卡片——如果它们与你的指控相符，你就赢得胜利啦！无需多说，你肯定希望在提出指控前有绝对的把握。

所以有了这道题。你是玩家 1。给定你手上的卡片集合，以及一些提议与反驳的记录，你需要确定你离能够做出指控还有多远。

### Input

第一行包含一个整数  $n$  ( $1 \leq n \leq 50$ )，表示游戏中已经完成的提议数量。

第二行包含你手上五张卡片的信息，它们是 'A' 到 'U' 之间的大写字母。

接下来  $n$  行，每行给出一条提议的信息。每行以三张卡片开始，依次表示提议的角色、武器和房间，接下来给出剩下三个玩家的响应，从提议者右边的玩家开始。如果一个玩家没有证据反驳，则响应信息是一个 '-'，否则响应信息是一个证据字符。如果证据对你可见的 (你是提议者或反驳者)，则证据字符是对应的卡片，否则证据字符是 '\*'。注意只有最后一条响应信息可以是证据字符。

所有字符之间用一个空格隔开。保证输入的提议和响应序列是合法的。

## Output

输出三个字符表示能识别出的凶手、凶器和作案地点。如果凶手可以被识别，输出相应的卡片字符，否则输出 '?'。凶器和作案地点同理。

## Examples

standard input	standard output
1 B I P C F A G M - - -	AGM
standard input	standard output
2 A B C D H F G M M F H M - *	E??
standard input	standard output
3 A C M S D B G S - G A H S - - S C J S *	???

## Problem C. Mission Improbable

Input file:           standard input  
Output file:          standard output  
Time limit:          1 second  
Memory limit:        1024 megabytes

那是春日里一个天气晴朗的好日子，你准备去见见你的老朋友 Patrick，也是你之前的犯罪同伙。Patrick 在编程竞赛上豪赌输掉了一大笔钱，所以他需要再干一票。为此他需要你的帮助，虽然你已经金盆洗手了。你刚开始很不情愿，因为你一点也不想再回到那条老路上了，但是你觉得听一下他的计划也无伤大雅。

在附近的一个仓库里有一批货物，包含一些贵重的消费性部件，Patrick 企图从中尽可能多地偷些东西出来。这意味着要找一条进去的路，弄晕安保人员，穿过各种各样的激光射线，你懂的，都是常见的抢劫技术。然而，仓库的核心装备了一套 Patrick 搞不定的安保系统。这也是他需要你帮助他的地方。

这批货物被放置在一些巨大的立方体箱里，每个箱子的尺寸都是相同的。这些箱子堆放成许多整齐的堆，每个箱子可以表示成一个三维的网格。安保系统每个小时会用三台相机对这堆货物进行一次拍照，相机分别为：前置相机 (front camera)，侧置相机 (side camera) 和顶置相机 (top camera)。前置相机的照片显示了每一行最高的那堆箱子的高度，侧置相机显示了每一列最高的那堆箱子的高度，顶置相机显示了每个位置是否存在一堆箱子。如果安保系统发现任何一张照片出现了变化，它会立即拉响警报。

一旦 Patrick 进去了，他会确定每堆箱子的高度并且发给你。图 1 显示了一种网格可能的放置，以及每台相机会得到的视图。

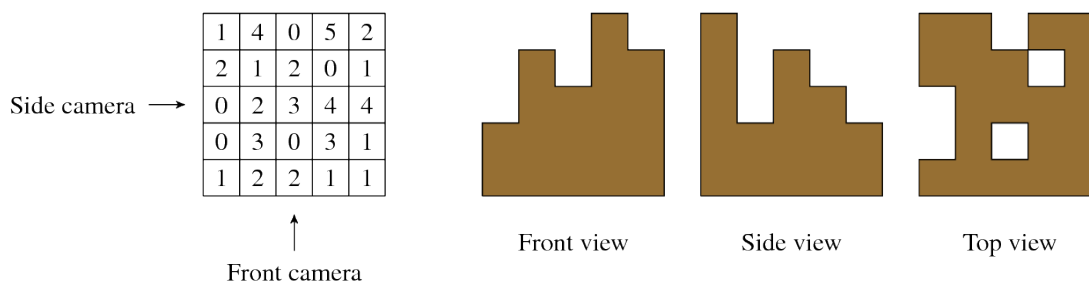


图 1. 网格的高度值与对应的相机视图。

1	4	0	5	1
2	1	1	0	1
0	1	3	1	4
0	3	0	1	1
2	1	1	1	1

图 2. 洗劫后网格可能的高度值。

Patrick 想尽可能多偷走一些箱子。由于他不能弄坏安保系统，他准备重新安排剩余每堆箱子的放置，使得下一次相机取像时会得到相同的照片，从而骗过安保系统。在上面的例子中，他可以偷走九个箱子。图 2 显示了一种可能的剩余箱子的安置方案能使得安保系统认为与原安置情况相同。

Patrick 想请你帮他确定在保证能骗过安保系统的情况下他最多能偷走多少个箱子。你会帮他干完这

最后一票么？

Input

第一行包含两个整数  $r$  ( $1 \leq r \leq 100$ ) 和  $c$  ( $1 \leq n \leq 100$ )，分别表示网格的行数与列数。

接下来  $r$  行，每行包含  $c$  个整数，表示对应行上每堆立方体箱的高度 (箱子的数量)。所有的高度在 0 到  $10^9$  之间 (含边界)。

Output

输出在不被发现的情况下最多能偷走多少箱子。

Examples

standard input	standard output
5 5 1 4 0 5 2 2 1 2 0 1 0 2 3 4 4 0 3 0 3 1 1 2 2 1 1	9
standard input	standard output
2 3 50 20 3 20 10 3	30

## Problem D. Money for Nothing

Input file:           standard input  
 Output file:         standard output  
 Time limit:          5 seconds  
 Memory limit:       1024 megabytes

在这道题种你需要解决一个全世界人类从存在起就在面临的最深刻的问题——如何发大财。

你是一名零件交易市场的中介。你的工作是从零件生产公司那里买到零件，然后把它们卖给零件消费公司。每个零件消费公司在截止日期前每天都会对一个零件有一个开放式的需求，以及它愿意买下零件的价格。另一方面，每个零件生产公司在开始日期及以后都可以销售零件，以及它销售零件的价格。

基于公平竞争法，你只能与一家生产公司、一家消费公司签订合同。你可以在生产公司开始销售后每天从生产公司买一个零件，当然这也要在消费公司结束需求之前。在这些天里，每天你可以从买卖差价中获取利润。

你的任务是选择能使你利益最大化的生产公司与消费公司。

### Input

第一行包含两个整数  $m$  和  $n$  ( $1 \leq m, n \leq 500\,000$ )，分别表示市场里生产公司与消费公司的数量。

接下来  $m$  行，第  $i$  行包含两个整数  $p_i$  和  $d_i$  ( $1 \leq p_i, d_i \leq 10^9$ )，表示第  $i$  个生产者卖一个零件的价格和第一个零件开始卖的日期。

接下来  $n$  行，第  $j$  行包含两个整数  $q_j$  和  $e_j$  ( $1 \leq q_j, e_j \leq 10^9$ )，表示第  $j$  个消费者愿意买一个零件的价格和它可以接收最后一个零件的日期的下一天。

### Output

输出你最多能赚到多少钱。如果你没办法通过签合同获利，输出 0。

### Examples

standard input	standard output
2 2 1 3 2 1 3 5 7 2	5
standard input	standard output
1 2 10 10 9 11 11 9	0

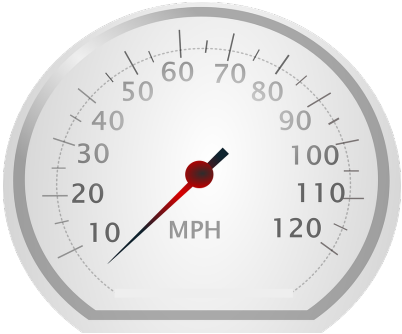
This page is intentionally left blank.

此页有意留为空白。



## Problem E. Need for Speed

Input file:           standard input  
Output file:          standard output  
Time limit:          1 second  
Memory limit:        1024 megabytes



Sheila 是一名学生，她开着一辆经典的学生车：一辆又老，又慢，又锈，还老是崩坏的车。最近，时速表盘的指针还掉了。她把指针粘了回去，但是她可能没有粘对角度。因此，当表盘读数为  $s$  时，她真实的速度可能是  $s + c$ ，其中  $c$  为未知常数（可能是负的）。

Sheila 在最近的行程中仔细地做了一些记录，并希望能用这些记录来计算出  $c$  的值。行程由  $n$  段组成。在第  $i$  段中，她匀速行驶了  $d_i$  的距离，表盘对应的读数一直为  $s_i$ 。整个行程花费的时间为  $t$ 。请你帮 Sheila 确定  $c$  的值。

注意即使 Sheila 的表盘可能有负的读数，她在每段行程的真实速度也是大于零的。

### Input

第一行包含两个整数  $n$  ( $1 \leq n \leq 1000$ ) 和  $t$  ( $1 \leq t \leq 10^6$ )，分别表示 Sheila 的行程段数和总时间。

接下来  $n$  行，每行描述了 Sheila 的一段行程。第  $i$  行包含两个整数  $d_i$  ( $1 \leq d_i \leq 1000$ ) 和  $s_i$  ( $|s_i| \leq 1000$ )，分别表示第  $i$  段行程的距离和表盘读数。

时间单位是小时，距离单位是英里，速度单位是英里每小时。

### Output

输出常数  $c$ ，其单位是英里每小时。你的答案绝对或相对误差应该小于  $10^{-6}$ 。

### Examples

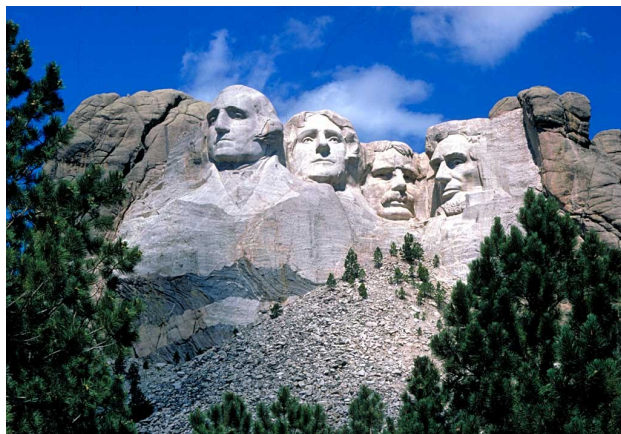
standard input	standard output
3 5 4 -1 4 0 10 3	3.000000000
standard input	standard output
4 10 5 3 2 2 3 6 3 1	-0.508653377

This page is intentionally left blank.

此页有意留为空白。

## Problem F. Posterize

Input file: standard input  
 Output file: standard output  
 Time limit: 2 seconds  
 Memory limit: 1024 megabytes



数字图像的像素可以用三个在 0 到 255 之间的整数表示，它们分别表示红色、绿色和蓝色的强度。为了压缩图片或是为了产生艺术效果，许多图像编辑工具收录了如下所述的“色调分离”操作。每个颜色通道会分别考虑，本题只考虑红色通道的情况。不同于在红色通道使用 0 到 255 之间全部的整数，一张色调分离后的图片只会使用这些数字里至多  $k$  种整数。每个像素原来的红色强度会被替换成最相近的可用强度。图像编辑工具会选择  $k$  个整数来最小化替换过程引起的平方误差之和。假设原图有  $n$  个像素，它们的红色取值是  $r_1, \dots, r_n$ ，而  $k$  种可用整数为  $v_1, \dots, v_k$ ，那么平方误差之和被定义为

$$\sum_{i=1}^n \min_{1 \leq j \leq k} (r_i - v_j)^2$$

你的任务是计算可以实现的最小平方误差之和，参数  $k$  和图片的红色强度会给出。

### Input

第一行包含两个整数  $d$  ( $1 \leq d \leq 256$ ) 和  $k$  ( $1 \leq k \leq d$ )，分别表示原图中不同的红色强度有多少种，色调分离后可以使用的红色强度有多少种。

接下来  $d$  行描述了每种红色强度在原图中占据的像素点数量。每行包含两个整数  $r$  ( $0 \leq r \leq 255$ ) 和  $p$  ( $1 \leq p \leq 2^{26}$ )，这里  $r$  是一种红色强度的取值，而  $p$  是这种取值对应的像素点数量。这  $d$  行信息按照红色强度取值升序给出。

### Output

输出最优的  $k$  种可选取值对应的平方误差之和。

**Examples**

standard input	standard output
2 1 50 20000 150 10000	66670000

standard input	standard output
2 2 50 20000 150 10000	0

standard input	standard output
4 2 0 30000 25 30000 50 30000 255 30000	37500000

## Problem G. Replicate Replicate Rfplicbte

Input file: standard input  
 Output file: standard output  
 Time limit: 3 seconds  
 Memory limit: 1024 megabytes

细胞自动制造公司 (Automatic Cellular Manufacturing) 最近刚刚获得批量生产零件的新工艺专利。它的方法使用到包含两种细胞状态的二维网格，每个单元的细胞要么为空，要么为满。当然，具体的细节是专有的。

最初，网格中的一组单元被填充为需要复制的细胞副本。经过一系列离散的步骤，网格中的每个单元会根据自身及附近的八个单元的状态同步进行细胞状态更新。如果这九个单元中有奇数个位置的细胞是满的，那么这个单元的下一个状态也是满的，否则会是空的。图 1 显示了一个由三个满细胞的单元组成的简单模式在复制过程中的几个步骤。

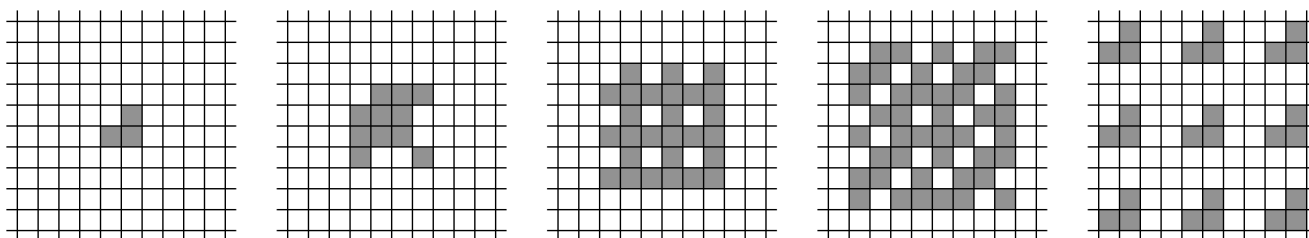


图 1. 复制过程。

然而，一个 bug 已经蛰伏在工艺中了。在每次更新单元状态之后，网格中某一个单元的状态可能自动变化。例如，图 2 显示了可能的变化，其中一个细胞在第一次状态更新后出现了自动变化，另一个细胞在第三次状态更新后出现了自动变化。

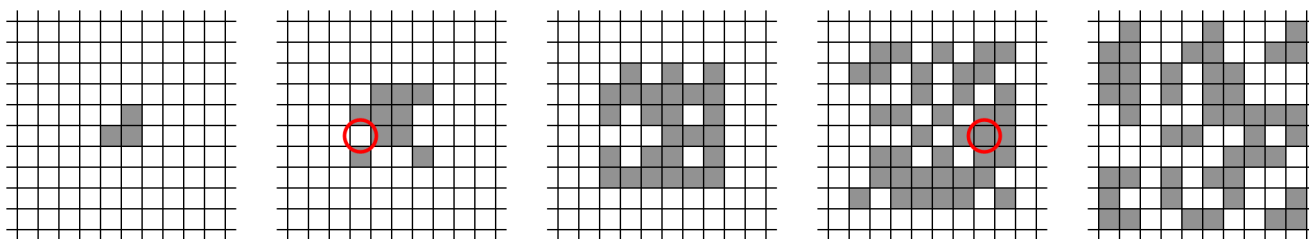


图 2. 复制过程中的错误。该图对应样例输入 1。

很不幸，最初的细胞模式丢失了，只有 (可能受到 bug 影响的) 复制结果保留了下来。你能否编个程序确定可能的最小的非空的初始细胞模式来产生给定的最终模式呢？

### Input

第一行包含两个整数  $w$  ( $1 \leq w \leq 300$ ) 和  $h$  ( $1 \leq h \leq 300$ )，其中  $w$  和  $h$  表示最终模式的边框宽度与高度。

接下来  $h$  行，每行包含  $w$  个字符，描述了最终的模式。每个字符要么是 '.' (表示该单元为空)，要么是 '#' (表示该单元为满)。

保证在第一行、最后一行、第一列、最后一列均存在至少一个单元是满的。

Output

输出最小的非空的可能产生最终模式的初始模式，这里假设每个阶段至多会有一个单元状态自动变化。模式的大小取决于边框内的区域。如果由多种可能的最小的非空的初始模式，那么任意一种答案都是可接受的。请你使用字符 '.' 表示空的单元，使用 '#' 表示满的单元。请你用必须要使用的最小行数和列数来输出这个模式。

Examples

standard input	standard output
10 10 . #...#...# ##...##...## ##.##... ##.##... . ...##### ...##...#.# .....###. ##.##... #...#...# ##...##...##	.# ##
standard input	standard output
8 8 ##...#. #.#####. .#.##... .##.##. .#.##... .##.##. #...#### ##.##.	#### #..# #..## ####.
standard input	standard output
5 4 #... ..### ..### ..###	#

## Problem H. Scenery

Input file:           standard input  
Output file:          standard output  
Time limit:           6 seconds  
Memory limit:        1024 megabytes



图片来自 John Fowler, Carol Highsmith, Richard Woodland

你决定在旅途中花一天的时间在 Rapid City 拍摄一些关于 South Dakota Badlands 的照片，这些地方以其壮观而不寻常的地层闻名。你是一名业余摄影师，但是对摄影的光照条件有着严苛的要求。

经过一番仔细的研究，你发现 Badlands 一处美丽的地方，周围环绕着如画般的风景。你已经确定了你想在这里拍摄的一系列特色。对于每个特色你也确定了一天中阳光最理想的最早时间与最晚时间。然而，你在拍照时需要花相当多的时间来重置三脚架与相机的位置，以满足你的完美主义。所以你想知道你是否可能在一天内成功完成所有特色的拍摄。

### Input

第一行包含两个整数  $n$  ( $1 \leq n \leq 10^4$ ) 和  $t$  ( $1 \leq t \leq 10^5$ )，其中  $n$  表示你想拍的照片数量， $t$  表示你拍每张照片要花的时间。

接下来  $n$  行，每行描述了一张照片理想的拍照时间段。每行包含两个非负整数  $a$  和  $b$ ，其中  $a$  是你开始拍摄这张照片的最早时间，而  $b$  是这张照片必须完成的时间，满足  $a + t \leq b \leq 10^9$ 。

### Output

如果可以拍这  $n$  张照片，输出 yes，否则输出 no。

### Examples

standard input	standard output
2 10 0 15 5 20	yes

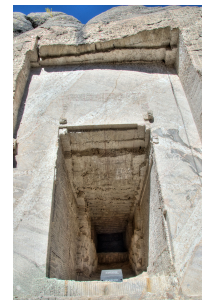
standard input	standard output
2 10 1 15 0 20	no

standard input	standard output
2 10 5 30 10 20	yes



## Problem I. Secret Chamber at Mount Rushmore

Input file: standard input  
Output file: standard output  
Time limit: 1 second  
Memory limit: 1024 megabytes



你现在可能已经听说过 Rushmore 山上有一个壮观的石雕刻画了四位著名的美国总统。然而，很少有人知道这个纪念雕刻暗藏一个秘密的房间。这听起来像是一部好莱坞电影的情节，但是这个房间是真实存在的。它藏在 Abraham Lincoln 的头后面，它被设计成一个档案库，用来存放美国历史上重要的文件与文物。历史学家声称这个档案库的建造在 1939 年被中断，直到 20 世纪 90 年代末才能被访问，但这不是全部的真相。

在 1982 年，著名的考古学家 S. Dakota Jones 秘密地访问了这座纪念雕刻，发现房间已经完成建造，但不对外公开。这似乎有些可疑，经过一些开凿，她发现一个隐藏的地下室和一些文件。不幸的是，这些文件没什么有效信息，都是很无聊的内容。她猜测这些内容被加密了，但是她费尽心机还是无法解密。

在这周的前些时候，Jones 博士在当地参加 ACM-ICPC 世界总决赛，她终于在 South Dakota School of Mines and Technology 的 Connolly Hall 发现了破译这些文件的密钥。她发现文件包含了一系列字母转换规则。一些字母可以经过多次转换，但一些字母没有转换规则。经过对那份无聊的文件中的单词多次转换之后，她似乎能把内容破译成美国历史上的文件，例如独立宣言和美国宪法。她需要你的帮助。

你将得到字母可能的转换规则和一系列原始单词与解密单词。你的任务是验证每对单词是否匹配。如果两个单词具有相同的长度，且第一个单词的每个字母可以用转换规则多次转换后变成第二个单词中对应位置的字母，则两个单词匹配。

### Input

第一行包含两个整数  $m$  ( $1 \leq m \leq 500$ ) 和  $n$  ( $1 \leq n \leq 50$ )，其中  $m$  表示字母转换规则的数量， $n$  表示单词对的数量。

接下来  $m$  行，每行包含两个不同的字母  $a$  和  $b$ ，表示字母  $a$  可以转换成字母  $b$ 。每个有序对  $(a, b)$  只会出现至多一次。

接下来  $n$  行，每行包含一对需要检查的单词。

转换规则和单词只会使用小写字母 'a' 到 'z'，且每个单词包含至少 1 个字母，至多 50 个字母。

### Output

对于每对单词，如果两个单词可以匹配，输出 yes，否则输出 no。

**Examples**

standard input	standard output
9 5 c t i r k p o c r o t e t f u h w p we we can the work people it of out the	yes no no yes yes

standard input	standard output
3 3 a c b a a b aaa abc abc aaa acm bcm	yes no yes

## Problem J. Son of Pipe Stream

Input file:           standard input  
 Output file:          standard output  
 Time limit:           5 seconds  
 Memory limit:        1024 megabytes

两年前，你在你的家乡协助安装了全国第一个 Flubber 管道网络并取得了巨大的成功。民意调查显示，每个人都喜欢在自己的厨房里安装他们自己的 Flubber 分配器 (类似水龙头)，现在有一些活跃的公民还发现了另一个用途。显然 Flubber 与水混合后有助于扑灭火灾！这是一个非常及时的发现，因为最近失控的火灾真是意外地常见。

你家乡的城市委员会希望在中心车站生产 Flubber 和水的混合物来充分利用 Flubber 的这个特性。这个被称为 Flubber Department (FD) 的车站也将配备训练有素的专业人员，他们负责前往火灾场所并利用加工过的 Flubber 来控制火情。

管道已经安置在整个城市之中。你需要通过管道的布局确定如何安排从 Flubber 工厂运送到 FD 的 Flubber 以及从当地水源到 FD 的水。

注意 Flubber 和水会流经相同的管道网络，甚至是同一个管道。所有管道都是双向的，但是 Flubber 和水不能在相同的管道中以相反的方向运输。此外，如果两种液体在相同的管道以相同的方向输送，它们将不可避免地混合。因此网络中的每个节点都配备了特殊的膜与过滤器，你可以根据自己的需要来分离和重组所有流进的混合物。网络是一个封闭的系统，所以除了来源地和目的地 (FD) 之外，流入每个节点的总流速必须等于流出的总流速。

每个管道都有固定的容量。Flubber 稍微粘稠一些，它具有粘度值  $v$ ，这意味着可以运输  $v$  升/秒的水的管道只能运输 1 升/秒的 Flubber。而管道的容量对于两者的混合物是呈线性分布的。准确来说，如果用  $c$  表示管道相对水的容量限制， $f$  和  $w$  表示流经管道的 Flubber 和水的速率 (均以升/秒计算)，则容量约束满足不等式  $v \cdot f + w \leq c$ 。

你主要关心的是制衡到达 FD 的混合物。你希望液体总量尽可能多，但你也需要足够的水来稀释 Flubber (未稀释时 Flubber 高度易燃)，并且需要足够的 Flubber (毕竟是 Flubber Department)！你已经想出了一个公式来衡量最终混合物的“价值”： $F^a \cdot W^{1-a}$ ，其中  $F$  是流入 Flubber 的速率，单位为升/秒， $W$  是流入水的速率，单位为升/秒， $a$  是给定的 0 和 1 之间的常数。

请你确定可以得到的  $F^a \cdot W^{1-a}$  最大值，以及如何安排 Flubber 和水来实现这个最大值。

### Input

第一行包含地点的数量  $n$  ( $3 \leq n \leq 200$ )，管道的数量  $p$  ( $n-1 \leq p \leq \frac{1}{2}n(n-1)$ )，实数值  $v$  ( $1 \leq v \leq 10$ ) 和  $a$  ( $0.01 \leq a \leq 0.99$ )。地点从 1 到  $n$  标号，Flubber 工厂是 1，水源是 2，FD 是 3。实数值在小数点后最多有 10 位数字。

接下来  $p$  行，每行描述一条管道。每行包含两个整数  $j$  和  $k$  ( $1 \leq j < k \leq n$ )，对应管道连接的两个地点，以及一个整数  $c$  ( $1 \leq c \leq 10$ )，对应管道的容量 (单位：升/秒)。

没有两条管道连接相同的两个位置。此外，保证管道网络是连通的。

## Output

首先，对于每条管道 (按输入的顺序)，输出两个值：Flubber 在其中的流速和水在其中的流速 (如果是从  $k$  流到  $j$ ，用负数表示)，使得  $F^a \cdot W^{1-a}$  最大。然后，输出这个最大值，精确到绝对误差不超过  $10^{-4}$ 。

如果存在多个解，那么任意一个解都是可接受的。所有的限制 (不能在同一管道中以不同的方向输送 Flubber 和水、流量平衡、管道容量、构造解与答案的一致性) 必须满足绝对误差不超过  $10^{-4}$ 。

## Examples

standard input	standard output
6 6 3.0 0.66	0.000000000 1.360000000
2 4 8	0.000000000 1.000000000
4 6 1	0.000000000 -1.000000000
3 6 1	0.000000000 0.360000000
4 5 5	0.880000000 0.000000000
1 5 7	-0.880000000 -0.360000000
3 5 3	1.02037965897

standard input	standard output
5 5 1.0 0.5	5 0
1 2 10	5 5
2 3 10	4.2 3.14159
3 4 10	4.2 3.14159
4 5 10	-4.2 -3.14159
3 5 10	5

## Problem K. Tarot Sham Boast

Input file:           standard input  
Output file:          standard output  
Time limit:           2 seconds  
Memory limit:        1024 megabytes

你的对手太坏了！在每年的年度石头剪刀布锦标赛上，你总是能进总决赛。（你的石头技巧无与伦比，你的布震撼人心！不过你的剪刀需要练练）但是每年，你的对手总是能打败你，即使他的操作看上去是完全是随机的！而且他发布新闻声称他无可匹敌。他的秘诀是什么？

幸运的是，你认为你已经找到了。就在今年，锦标赛之前的时候，你发现他拜访了城里的许多萨满。啊哈！原来他在使用超自然力量对付你！你觉得这次可以好好玩一玩了。所以你拜访了一群算命师，他们用塔罗牌帮你预测了对手在比赛中会使用的操作序列。

然而，你最初的兴奋已经过去了，你现在感觉自己有点智障。这怎么可能赢？最后你才发现你已经对这种随机预测的欺诈行为付出了许多钱。好吧，你可能会在比赛中留意其中一些预测。但是用哪些预测好呢？

在总决赛中，你和你的对手将玩  $n$  轮石头剪刀布。每一轮中，你和你的对手会从三种选择（石头、剪刀、布）中选择一个操作。根据你们的选择来确定每一轮的胜利者（实际上与本题没太大关系）。

给定总决赛的长度和一些预测，请你按照在总决赛中作为你对手操作的一个连续子序列出现的可能性排序它们，这里假设对方每一轮的选择是独立随机的。

### Input

第一行包含两个整数  $n$  ( $1 \leq n \leq 10^6$ ) 和  $s$  ( $1 \leq s \leq 10$ )，分别表示总决赛的轮数和序列的数量。

接下来  $s$  行，每行描述一种预测，包含一个由 'R', 'P', 'S' 组成的字符串。所有的预测拥有相同的长度，长度在 1 到  $n$  之间（含边界），且不会超过  $10^5$ 。

### Output

输出按照在总决赛出现的可能性降序输出所有的预测。对于可能性相同的预测，按照他们在输入中的顺序输出。

### Examples

standard input	standard output
3 4	PS
PP	PP
RR	RR
PS	SS
SS	

standard input	standard output
20 3	PRSPS
PRSPS	PPSPP
SSSSS	SSSSS
PPSPP	

## Problem L. Visual Python++

Input file:           standard input  
 Output file:         standard output  
 Time limit:          5 seconds  
 Memory limit:       1024 megabytes

在最近被提出的 Visual Python++ 编程语言中，一个语句块被表示为一个由字符组成的矩形，其中左上角在  $r_1$  行  $c_1$  列，右下角在  $r_2$  行  $c_2$  列。对于  $r_1 \leq r \leq r_2, c_1 \leq c \leq c_2$ ，所有位于  $(r, c)$  的字符被认为是属于这个块的内容。在这些位置中，满足  $r = r_1$  或  $r = r_2$  或  $c = c_1$  或  $c = c_2$  的位置被称为是边界。

语句块可以嵌套 (矩形包含在其他矩形中) 任意层。在语法正确的程序中，任意两个语句块要么是嵌套的 (一个包含在另一个中)，要么是不交的 (不重叠)。在这两种情况中，他们的边界也不能重叠。

编程人员不需要画出经典程序中的所有矩形，这太浪费时间了，而且 Visual Python++ 也不可能称为下一个 ICPC 编程语言。因此程序员只需要在左上角位置放一个字符 '┌'，在右下角位置放一个字符 '└'。解析器会自动匹配相应的拐角来获取程序的嵌套结构。

你的团队刚刚获得了五小时的合同来开发解析器的这一部分。

### Input

第一行包含一个整数  $n$  ( $1 \leq n \leq 10^5$ )，表示拐角对的数量。

接下来  $n$  行，每行包含两个整数  $r$  和  $c$  ( $1 \leq r, c \leq 10^9$ )，指定  $r$  行  $c$  列为一个左上角。

接下来  $n$  行以相同的方式指定了右下角。

所有的拐角位置互不相同。

### Output

输出  $n$  行，每行包含一个整数。第  $i$  行的整数  $j$  表示第  $i$  个左上角和第  $j$  个右下角组成一个矩形。左上角和右下角均按照他们在输入中的顺序从 1 到  $n$  标号。输出必须是 1 到  $n$  的排列，从而匹配可能嵌套的矩形。如果存在超过一种合法的匹配，任意一组合法的匹配都是可接受的。如果不存在合法的匹配，输出 syntax error。

### Examples

standard input	standard output
2	2
4 7	1
9 8	
14 17	
19 18	

standard input	standard output
2 4 7 14 17 9 8 19 18	1 2

standard input	standard output
2 4 8 9 7 14 18 19 17	syntax error

standard input	standard output
3 1 1 4 8 8 4 10 6 6 10 10 10	syntax error