



问题A. Y形刀

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

围绕樱桃巧克力蛋糕经常会引起激烈的争论。每个人都喜欢樱桃，所以蛋糕必须以每个人都收到相同数量樱桃的方式进行切割。

英石

两个人只需用一把简单的刀直切即可轻松分割蛋糕。虽然不太明显，但两次直切对于四个人来说总是足够的。但是如果你想切三个人的蛋糕怎么办？我们有解决方案！

你会得到一个樱桃蛋糕和一把 Y 形刀。将蛋糕信息切成三部分，并放入相同数量的樱桃，否则报告这是不可能的。

正式地，您将获得平面上一般位置的一组点。Y形刀是一个顶点和从顶点出发的三条无限直射线。任意两条射线之间的角度等于 120° ($2\pi/3$)。刀将平面分成三部分。您必须找到顶点的位置和第一条射线的旋转，以使每个部分包含相同数量的点。

输入

输入的第一行有一个整数 n ，即点数 ($1 \leq n \leq 10^5$)。接下来的 n 行每行包含两个整数，即对应点的 x 和 y 坐标。坐标的绝对值不超过 10^6 。

保证没有两点重合，也没有三点在同一条直线上。

输出

如果无法进行所需的切割，请打印“否”（不带引号）。

如果可能，请打印“是”（不带引号）。然后打印三个实数 x 、 y 和 r ($|x|, |y| \leq 10^7$ 、 $0 \leq r < 2\pi/3$)，其中 x 和 y 表示顶点坐标， r 表示刀逆时针旋转的弧度。

刀的初始位置（即，对应于零旋转）是当其中一条射线指向下方时。射线指向右侧时的位置对应于 $\pi/2$ 的旋转。

有关检查器的详细说明，请参阅“注释”部分。

例子

standard input	standard output	illustration
9 3 -2 -4 6 0 -7 -5 -6 5 1 1 6 -5 0 -3 -7 -4 2	Yes -1 0 0.9	

笔记

所有计算均以 long double (80 位浮点类型) 完成。设顶点为 (x, y) , 旋转为 r 。对于每个输入点 (x_i, y_i) :

- 如果 $|x - x_i| < 10^{-9}$ 和 $|y - y_i| < 10^{-9}$, 则解得到“错误答案”;
- 从顶点到输入点的方向角度计算为 $\text{atan2}(y_i - y, x_i - x) + \pi/2 - r$ modulo 2π , 其中 $\text{atan2}(y, x)$ 是点 (x, y) 的极角;
- 如果角度与 0 、 $2\pi/3$ 或 $4\pi/3$ 的绝对差小于 10^{-9} , 则解得到“错误答案”;
- 最后直接计算出该点所属的区间。



问题 B. 坏医生

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

亚历克斯生病了。他去了一家诊所并拜访了 n 医生。第 i 位医生说，从 l_i 天开始到 r_i 天结束，Alex 必须服用 k_i 种药物: a_1, a_2, \dots, a_{k_i} ，每天一粒。药品编号从 1 到 m 。

当然，如果几个医生告诉亚历克斯在同一天服用同一种药，那么他当天只会服用这种药的一粒。至少，人们在现实生活中是这样的。

一粒药 j 的价格为 c_j 卢布。但亚历克斯有一个疑问：谣言说诊所的一位医生确实很糟糕。他不知道哪个医生不好，但他决定忽略这个医生的处方。

你的任务是找到 n 个数字 t_i : 知道如果第 i 个医生不好，亚历克斯会花多少钱买药。

输入

第一行包含两个整数 n 和 m : 医生数量和药品数量
($1 \leq n \leq 500\,000$, $1 \leq m \leq 500\,000$)。

t 第二行包含 m 整数 c_j : 第 j 种药物的一粒药的成本 ($1 \leq c_j \leq 1\,000\,000$)。

接下来的每一行 n 都描述了医生。其中第 i 个以三个整数 l_i, r_i, k_i 开头：第 i 个医生处方的开始和结束日期以及他告诉 Alex 服用的药物数量 ($1 \leq l_i \leq r_i \leq 1\,000\,000$, $1 \leq k_i \leq m$)。然后跟随 k_i 不同的整数 a_1, a_2, \dots, a_{k_i} ，每个从 1 到 m : 处方中的药物。

输入中所有 k_i 的总和不超过 $1\,000\,000$ 。

输出

输出 n 个整数 t_1, t_2, \dots, t_n : 如果 Alex 忽略第 i 个医生的处方，他将花多少钱买药。

例子

standard input	standard output
5 4 1000 100 10 1 3 4 2 2 3 4 8 3 1 2 4 6 7 2 3 4 8 9 2 1 4 2 6 3 1 2 3	8766 7564 8756 7765 6646



问题 C. 拓扑排序

输入文件: 标准输入 输出文件: 标准
输出时间限制: 4 秒 内存限制: 512
MB

有向无环图的拓扑排序是其顶点 p_1, \dots, p_n 的排列。对于每个弧，在排列中其源位于其目标之前。

在

给你一个有向无环图。对于每对顶点 (u, v) , 计算拓扑排序的数量，使得顶点 u 位于顶点 v 之前。

输入

第一行包含一个整数 t , 即测试用例的数量。 t 测试用例的描述如下。

每个测试用例的第一行有两个整数 n 和 m : 顶点和弧的数量 ($1 \leq n \leq 20, 0 \leq m \leq n \cdot (n - 1)/2$)。

接下来的 m 行包含两个整数 u_i 和 v_i , 表示从顶点 u_i 到顶点 v_i ($1 \leq u_i < v_i \leq n$) 的弧。

有在莫输入中的 st 100 个测试用例。最多 5 个测试用例 $n > 10$.

输出

对于每个测试用例，打印 n 行，每行包含 n 数字。第 i 行中的第 j 个数字应等于顶点 j 在顶点 i 之前的拓扑排序数。特别是，如果 $i = j$ 它应该等于 0。

例子

standard input	standard output
2	0 0 0
3 2	2 0 1
1 2	2 1 0
1 3	0 0 3 1
4 2	6 0 5 3
1 2	3 1 0 0
3 4	5 3 6 0



问题 D. 括号欧拉游览

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

给您一个具有 n 个顶点和 m 个边的无向图。每个顶点都有一个与其关联的括号（左 “(” 或右 “)” ）。

您必须在该图中找到一条欧拉之旅，使其顶点（按遍历顺序编写）形成正确的括号序列。

正确的括号序列是通过在序列的原始字符之间插入字符 “1” 和 “+” 可以将括号序列转化为正确的算术表达式。例如，括号序列 “()0” 、 “(0)” 是正确的（结果表达式是 “(1)+(1)” 和 “((1+1)+1)” ），而 “)(” 和 “0(” 则不正确。

无向图的欧拉巡演是一个循环，该循环仅访问该图中的每条边一次。不过，允许多次访问同一顶点。

输入

第一行包含两个整数 n 和 m ($1 \leq n, m \leq 2 \cdot 10^5$)，分别是顶点数和边数。

以下 m 行中的每一行都包含两个整数 v_i 和 u_i ($1 \leq v_i, u_i \leq n$)，这意味着顶点 v_i 和 u_i 之间有一条无向边。请注意，允许自循环和多条边。

输入的最后一行包含一串 n 圆括号，其中第 i 个括号与顶点 i 关联。

输出

如果给定图中不存在构成正确括号序列的欧拉游览，则在第一行打印 “No” 。

否则，在第一行打印 “Yes” 。在第二行中，打印形成欧拉环路的顶点序列以及正确的括号序列。如果有多个解决方案，请打印其中一个。

例子

standard input	standard output
2 2 1 2 1 2)(Yes 2 1
5 6 1 2 2 3 3 1 2 4 4 5 5 2 (())	Yes 1 2 4 5 2 3
1 1 1 1 (No



问题 E. 电荷树

输入文件: 标准输入
输出文件: 标准
输出时间限制: 4 秒
内存限制: 512 MB

有一棵有根的树。每个顶点都包含一些非负电荷量 c_v 。你必须继续
三种查询:

SS

- 向上移动电荷: 所有顶点同时将其所有电荷转移到其直接父节点。来自根部的电荷不会转移到任何地方。也就是说, 如果顶点 v 具有子节点 u_1, \dots, u_k , 则对于非根顶点 v , 其新电荷变为 $c_{u_1} + \dots + c_{u_k}$; 如果 v 是根, 则其新电荷变为 $c_{u_1} + \dots + c_{u_k} + c_v$
- 向下移动电荷: 所有顶点同时将其所有电荷以相等的比例转移给其子节点。也就是说, 如果顶点 v 具有子节点 u_1, \dots, u_k , 则每个 u_i 的新电荷变为 c_v/k 。每片叶子上都附有一个足够长度的虚拟线树。也就是说, 如果电荷从叶子向下移动, 然后向上移动相同的次数, 则叶子保留其原始电荷。
- 向顶点添加电荷: 向某个顶点添加一定量的电荷。

最后, 您应该打印每个顶点的电荷值。

输入

输入的第一行有一个整数 n , 即树中的顶点数 ($2 \leq n \leq 500\,000$)

下一行包含 n 个整数 c_i , 其中 i 表示树的初始电荷 ($0 \leq c_i < 10^9 + 7$)。

接下来的每行 $n - 1$ 包含两个整数 u_i 和 v_i 表示顶点 u_i 和 v_i 之间的边
($1 \leq u_i, v_i \leq n$)。

下一行包含一个整数 q , 即查询数量 ($0 \leq q \leq 500\,000$)。然后 q 行后面是查询的描述。“上移”查询用字符“ \gg ”表示。“下移”查询用字符“ \ll ”表示。“添加电荷”查询用字符“ $+$ ”表示, 后跟两个整数 v_i 和 x_i , 这意味着您应该将电荷 x_i 添加到顶点 v_i ($1 \leq v_i \leq n$, $0 \leq x_i < 10^9 + 7$)。

保证输入中的图是一棵树。

输出

打印 n 个数字, 其中第 i 个数字是顶点 i 模 $10^9 + 7$ 的最终电荷。

正式地, 令电荷为 p/q 。然后你应该打印一个唯一的数字 x , $0 \leq x < 10^9 + 7$, 这样 $p \equiv x \cdot q \pmod{10^9 + 7}$ 。



例子

standard input	standard output
5 4 3 3 6 0 1 2 1 3 2 4 4 5 5 v + 1 7 ~ + 2 1 v	0 500000009 500000009 4 6
2 5 10 1 2 5 v v v ~ ~	0 5
4 0 1 0 0 1 2 1 3 1 4 2 ~ v	0 33333336 33333336 33333336



问题 F. 找到一棵树

输入文件: 标准输入 输出文件: 标准
输出时间限制: 4 秒 内存限制: 512
MB

无向图 $G(V, E)$ 的正确 k 着色是映射 $c: V \rightarrow \{1, 2, 3, \dots, k\}$, 这样对于每条边 $(u, v) \in E$, 我们有 $c_u \neq c_v$

如果存在适当的 k 着色, 则无向图是可 k 着色的。

图的色数是最小的 k , 使得图是 k 可着色的。

树是一个简单的无环无向图。

Alice 有一个色数为 k 的无向图, Bob 有一个位于 k 顶点的树。Bob 希望在 Alice 的图中找到 k 个不同的顶点 $p_1, p_2, p_3, \dots, p_k$, 这样对于 Bob 树中的每条边 (u, v) , Alice 的图中都存在一条边 (p_u, p_v) 。帮助他。

输入

第一行包含一个整数 T ($1 \leq T \leq 10^6$), 即要解决的测试用例的数量。 T 测试用例的描述如下。每个测试用例描述如下。

第一行包含三个整数 n 、 m 和 k ($1 \leq n, k \leq 10^6$ 、 $0 \leq m \leq 10^6$), 分别表示 Alice 图的顶点数和边数及其色数。

接下来的每一行 m 都包含一对整数 u_i 和 v_i ($1 \leq u_i, v_i \leq n, u_i \neq v_i$), 描述 Alice 图的一条边。保证不存在重边, 并且 Alice 的图的色数恰好等于 k 。

接下来的 $k - 1$ 行包含一对整数 p_i 和 q_i ($1 \leq p_i, q_i \leq k$, $p_i \neq q_i$) 描述
鲍勃树的边缘。保证给定的边集形成一棵树。

保证所有测试用例中的 n 之和以及所有测试用例中的 m 之和不为 n 超过 10^6 。
OT

输出

对于每个测试用例, 以以下格式输出答案。

如果无法在 Alice 的图中找到所需的 k 顶点, 则打印 “No”。

否则, 在第一行打印 “Yes”。在第二行中, 打印 k 个不同的整数 p_i ($1 \leq p_i \leq n$): Alice 图中对应于 Bob 树的各个顶点的顶点数。如果有多个可能的答案, 请打印其中之一。



例子

standard input	standard output
3	Yes
6 6 3	3 2 1
1 2	Yes
2 3	4 1 2 3
3 1	Yes
1 4	5 4 3
2 5	
3 6	
1 2	
2 3	
4 6 4	
1 2	
1 3	
1 4	
2 3	
2 4	
3 4	
1 2	
1 3	
1 4	
5 4 3	
1 2	
3 4	
4 5	
5 3	
1 2	
2 3	



问题 G. 一个根

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

对于固定的 n 和整数 p 和 q , 使得 $|p| \leq m$ 和 $|q| \leq m$, 有多少个以下形式的方程

$$x^n + px + q = 0$$

是否只有一个实根?

输入

输入的唯一行包含两个整数 n 和 m ($1 \leq n, m \leq 10^6$ 、 $n \geq 2$)。

输出

打印单张 teger: 恰好具有 1 个 r 的方程的数量 真正的根。

例子

standard input	standard output
2 4	5
3 5	96



问题 H. 删除点

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

平面上有 n 个点。点的坐标是整数。数字 n 始终是偶数。

您可以绘制一个边长与坐标轴平行的正方形。它的顶点可以有真实的坐标。如果正方形内部或其边界上正好有两个点，则将它们删除。

你必须找到一种方法来删除所有点，或者说这是不可能的。

输入

第一行包含一个整数 n ($1 \leq n \leq 3000$)，即点数。

接下来的每行 n 包含两个整数 x_i 和 y_i ($0 \leq x_i, y_i \leq 10^9$)，描述第 i 点的坐标。

所有点都是不同的。

输出

如果无法删除所有点，请在第一行打印“否”。

否则，在第一行打印“Yes”。在接下来的 $n/2$ 行中，打印四个实数：正方形对角的坐标。正方形应按照绘制的顺序打印。如果有多个可能的答案，请打印其中之一。

输出实数时小数点后不超过四位数。按以下形式打印它们：可能是一元减号，然后是任意数量的十进制数字，然后可能是一个小数点，后跟最多四个十进制数字。

例子

standard input	standard output
4 1 1 2 2 5 5 6 6	Yes 1.0 1.0 2.0 2.0 5.0 5.0 6.0 6.0
4 0 0 1 2 2 1 4 4	Yes 1.0 1.0 2.0 2.0 0.0 0.0 4.0 4.0



问题 I. 您的数据面临困难

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

有一个 n 节点的存储集群。每个文档恰好存储在两个节点中的两个副本中。

艰难的时刻即将到来。预算正在缩减。存储位变得非常昂贵，因此需要进行一些优化。确定每个节点的最大容量；即其中可以存储多少个副本。

由于现有副本无法移动，我们故意决定从集群中删除一些文档以满足新的需求。我们希望完全利用集群，因此在清理之后，每个节点应该存储其容量允许的尽可能多的文档。

我们以某种方式知道可以删除一些文档，以便完美地利用每个节点。然而，事实证明很难找到要删除的确切文档。你能帮助我们吗？

输入

第一行输入有两个整数 n 和 m ($1 \leq n \leq 500, 0 \leq m \leq n \cdot (n - 1)/2$)，
节点数和文档集的数量。

嗯

下一行包含 n 个整数 f_1, \dots, f_n ($0 \leq f_i \leq 10^9$)，其中 f_i 表示节点 i 的容量。

接下来的每行 m 包含三个整数 u_i, v_i, c_i ($1 \leq u_i, v_i \leq n, u_i \neq v_i, 1 \leq c_i \leq 10^4$)。这
描述意味着节点 u_i 和 v_i 中有带有副本的 c_i 文档。

e

不同文档集的描述中的所有节点对都是不同的。

输出

首先，打印一个数字 k ：删除后具有公共副本的节点对的数量。

es

然后打印 k 行，每行包含三个整数，格式与输入相同：点头对
以及这些节点中存储的文档数量。

删除之后，每对节点不能拥有比之前更多的文档。此外，
第 i 个节点中存储的文档数量必须等于 f_{i0} 。

e

保证答案存在。如果有多个答案，请打印其中之一。

例子

standard input	standard output
5 6	3
3 1 1 1 2	2 3 1
1 2 1	1 4 1
1 4 1	1 5 2
2 3 2	
2 4 1	
3 4 1	
1 5 3	



问题 J. 程序

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

给定一个使用整数变量 X 运行的程序，该变量最初等于 1。该程序由两种类型的 n 指令组成：

- “1 p” ($1 \leq p \leq n$)，将值 p 赋给变量 X 。
- “2 p q” ($1 \leq p, q \leq n, p \neq q$)，仅当 X 的当前值为 p 时，才将值 q 赋给变量 X 。

您只需一步即可从程序中删除任何一条指令。您无法重新排序说明或添加新说明。要获得这样一个程序，在运行后变量 X 具有值 k 所需的最少步骤是多少？要求您针对从 1 到 n 的每个 k 解决此问题。

输入

输入的第一行包含一个整数 n ($2 \leq n \leq 10^6$)，即指令的数量

程序中的蒸发散。

接下来的 n 行包含上述格式的指令说明。

输出

输出 n 个整数，其中第 i 个整数是使程序设定值所需的最小步数

i 变量 X ，或 -1（如果不可能）。

例子

standard input	standard output
3 1 1 1 2 1 3	2 1 0
4 2 1 2 1 3 2 2 3 2 3 1	0 2 1 -1