

## 问题 A.古老的地图，隐藏的危险

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
	1024 兆字节

看哪，冒险家！不要踏入深渊。

传说中，迷宫里总会有牛头怪。但迷宫的定义是什么呢？无数悲惨事件后，伟大的勇士艾伦决定亲自动手。

作为镇上有名望的战士之一，有一天，他根据自己以前的经验提出了一些猜想。当你只能呆在建筑物外面时，眼睛看不到的地方可能才是真正的危险源。他召集了酒吧里的朋友讨论这个新想法。，他们得出了如何科学测量这一特性的结论。

请在脑海中想象一幅二维结构图。该结构由多面墙组成，可以用多边形来描述。这些墙壁阻挡了移动和视线。也就是说，你无法行走，也无法透过墙看东西。墙壁上没有洞（艾伦是一名战士，而不是法师），因此，如果一个点不在墙壁里，就可以从外面到达。讨论的主要启示是，如果某个可到达的点在建筑外是可见的，那么对于任何距离来说，它都应该在与该点相同距离的某个地方以某个角度可见。从数学上讲，对于任意半径  $r$ ，我们都可以在以该点为圆心的圆上选取一个点，使得该点与该点之间的线段不与任何墙壁相交。任何在外面看不到的点都被称为**隐蔽点**，它们都在可到达但隐蔽的区域内。



图 A.1：样本输入 1 的迷宫地图。墙壁阴影为棕色，可到达但隐蔽的区域阴影为灰色。

当天晚上，乡亲们都困惑地盯着地图，进展。除了去实地考察，他们谁也没有测量区域的好方法。当然，这样做太冒险，也不切实际，因为艾伦不可能同时看守所有的人。也许你能帮忙？帮他们找到地图上隐藏的可到达区域，验证他们的猜想。

## 输入

第一行包含一个数字  $n$  ( $1 \leq n \leq 30$ )，即描述结构中墙壁的多边形的数目。下面  $n$  行中的每一行都包含几个整数。每行的第一个数字是第  $i$  个多边形的顶点数  $k_i$  ( $3 \leq k_i \leq 90$ )。接下来是  $2k_{(i)}$  个整数

、

$x_1, y_1, x_2, y_2 \dots, x_{k_i}, y_{k_i}$  ( $-10^4 \leq x_j, y_j \leq 10^4$ )，它们是多边形顶点的坐标、

按顺时针或逆时针顺序排列。输入中不会出现重复的顶点。给定的多边形是简单的，即没有洞且永远不会与自己相交。两个多边形不会在任何一点重叠。顶点总数不超过 90 个。

## 输出

输出一个浮点数，表示地图中可到达的隐藏区域。您的答案应

绝对或相对误差不超过  $10^{-6}$ 。

实例

标准输入	标准输出
2 8 0 0 7 0 7 7 6 7 6 1 1 1 1 7 0 7 4 2 7 5 7 5 6 2 6	2.25
2 4 2 4 5 8 9 4 5 7 4 4 5 5 1 6 5 5 4	1.4

## 问题 B. 基本图算法

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：1024 兆字节

Aqua 正在学习图论。在学习了深度优先搜索后，他编写了以下伪代码：

函数 dfs(u)

```

  将 u 标记为已访问
  过的输出 u
  对于 u 相邻顶点中的 v，如
    果 v 未被访问，则
      dfs(v)
    Endif
  结束功能

```

函数 run\_dfs()

```

  对于所有顶点中的 v
    如果 v 未被访问，则
      dfs(v)
    Endif
  Endfor

```

功能结束

Aqua 注意到第 4 行和第 12 行存在一些非确定行为。具体来说，遍历顶点的顺序可能不固定！

因此，Aqua 向您提出以下问题：给定一个有  $n$  个顶点和  $m$  条边的无向图，以及一个排列顺序  $p_1, p_2, \dots, p_n$ ，要使输出顺序成为排列  $p$ ，您需要添加的边的最少数量是多少？此外，您需要提供添加的边，以便 Aqua 检查您的答案。

可以证明，水叮当的问题总是有答案的。

### 输入

输入的第一行包含两个整数  $n, m$  ( $1 \leq n \leq 3 \times 10^5, 0 \leq m \leq 5 \times 10^5$ )，表示顶点和边的数量。

$m$  行，其中第  $i$  行包含两个整数  $u_i, v_{(i)}$  ( $1 \leq u_i, v_{(i)} \leq n, u_{(i)} \neq v_{(i)}$ )，表示第  $i$  条边连接顶点  $u_{(i)}$  和顶点  $v_{(i)}$ 。

最后一行包含  $n$  个整数  $p_1, p_2, \dots, p_n$ ，表示给定的排列组合。

## 输出

在第一行输出整数  $k$ ，即需要添加的最小边数。

后面有  $k$  行，其中第  $i$  行包含两个整数  $a_{(i)}$ 、 $b_{(i)}$ ，表示要添加的连接  $a_{(i)}$ 、 $b_{(i)}$  的第  $i$  条边 ( $1 \leq a_{(i)}, b_{(i)} \leq n, a_{(i)} \neq b_{(i)}$ )。

如果有多个答案，您可以输出任意答案。

实例

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

问题 c. 征服倍数

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 1024 兆字节

爱丽丝和鲍勃是两位杰出的战略家，他们在一座古老的寺庙里偶然发现了一块神秘的黑板，上面写着一串整数  $l, l+1, \dots, r$  的整数序列。黑板旁边有一个特殊的数字  $x$ ，初始设置为  $l$ 。

游戏规则很简单，但很有挑战性：爱丽丝和鲍勃轮流从黑板上恰好擦去一个  $x$  的倍数，爱丽丝总是先擦。每次移动后， $x$  的值会增加 1，游戏继续进行。如果一方无法擦除当前  $x$  的倍数，则输掉游戏。

两位棋手都渴望证明自己的战术才华，他们将以最佳的方式赢得胜利。您能确定哪位棋手将最终赢得这场多子之战吗？

输入

该问题包含多个测试用例。第一行输入包含一个整数  $T$  ( $1 \leq T \leq 10^5$ )，表示测试用例的数量。

每个测试用例只有一行输入。这一行包含两个整数  $l, r$  ( $1 \leq l \leq r \leq 10^9$ )，表示黑板上最初的数字范围 ( $l, l+1, \dots, r$ )。

输出

每个测试案例输出一行。如果 Alice 将获胜，则在这一行中打印 "Alice"。否则，打印 "Bob"。

示例

标准输入	标准输出
3	爱丽丝
2 4	爱丽丝
4 4	鲍勃
6 7	

问题 D. 减少和交换

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
	1024 兆字节

David Liu 的一行中有  $n$  个方格，第  $i$  个方格表示从左边开始的第  $i$  个方格，索引从 1 到  $n$ 。

弗兰克有一个长度为  $n$  的二进制字符串  $s$ 。他将在  $s_i=1$  的每个  $i$  中放  $10^{18}$  块石头，在  $s_{(i)}=0$  的每个  $i$  中放 0 块石头。

，David Liu 将尝试取出所有箱子中的每一块石头。但是，他不能直接把它们取出来，他只能通过执行以下操作 0 次或更多次才能实现这一目标：

- 选择一个整数  $i$ ，使得  $1 \leq i \leq n-2$ ，并且  $i$  箱中至少有一块石头。那么他必须从  $i$  箱中取出一块石头，并交换  $i$  箱+ 1 和  $i$  箱+ 2 中的物品。

由于石头很多，在自己尝试之前，David Liu 想问你是否有可能取出所有盒子里的每一块石头。

回顾一下，二进制字符串是指只包含 0 和 1 的字符串。

输入

第一行包含一个整数  $T$  ( $1 \leq T \leq 10^6$ )，代表测试用例的数量。每个测试用例有两行输入。

每个测试用例的第一行都包含一个整数  $n$  ( $3 \leq n \leq 10^6$ )，表示方框的数量。

每个测试用例的第二行包含一个长度为  $n$  的二进制字符串，表示弗兰克是否会在每个盒子里放石头。

保证所有测试用例的  $n$  之和不超过  $10^6$ 。

输出

对于每个测试用例，如果刘大卫能移除所有石头，则输出 "是"，否则输出 "否"。

示例

标准输入	标准输出
3	没有
3	是
101	是



4 1010 5 00000	
-------------------------	--

## 问题 E. 平等衡量

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
	1024 兆字节

一位朋友买了一块布料，绿底白格。"啊，看起来像棋盘！""对我来说，它看起来更像稿纸""真让我想起了绿豆糕"同样的布料，我们每个人的看法却不尽相同。每个人对美的看法都不尽相同，但有不同的看法并没有什么坏处。真正重要的是包容和尊重的观点。

如果一个人可以隔着门欣赏日出之美，为什么非要走到窗前听鸟鸣呢？你欣赏你的鸟鸣，他们欣赏他们的日出，双方都会感受到同样的美。

您决定去植物园欣赏鸟鸣。这里有  $n$  个区域和植物园中的  $m$  条步道，以及连接  $u_i$ -th 和  $v_i$ -th 区域的第  $i$  条步道。

植物园中的一条路线定义为长度为  $k$  的  $a_1, a_2, \dots, a_k$ ，长度为  $k$ ，并满足以下条件：

- $k \geq 3$ .
- $\forall i \neq j, a_i \neq a_j$ .
- $\forall 1 \leq i < k$ ，有一条小路连接  $a_i$ -th 区域和  $a_{i+1}$ -th 区域。
- 有一条小路连接  $a_k$ -th 和  $a_1$ -th 区域。

为了确保每位游客都能体验到同样美，你决定编写一个程序来检查植物园中的所有路线是否都一样长。如果植物园中不存在任何路线，我们就认为所有路线长度相同。

## 输入

第一行包含一个整数  $T$  ( $1 \leq T \leq 5 \times 10^5$ )，表示测试用例的数量。

在每个测试用例中，第一行包含两个整数  $n, m$  ( $1 \leq n \leq 5 \times 10^5, 0 \leq m \leq 5 \times 10^5$ )，分别代表区域和路径的数量。保证在所有  $T$  个测试用例中

$$\sum_n \sum_{m \leq 5 \times 10^5} m \leq 5 \times 10^5。$$

接下来是  $m$  行。其中第  $i$  条线包含两个整数  $u_i, v_i$  ( $1 \leq u_i, v_i \leq n, u_i \neq v_i$ )，表示第  $i$  条路径连接了第  $u_i$  个区域和第  $v_i$  个区域。最多有一条路径连接同一对区域。从任何一个区域，都可以通过小路到达任何其他区域。

## 输出

对于每个测试用例，如果所有路由长度相同，则输出 "是"，否则输出 "否"。

实例

标准输入	标准输出
1 5 6 1 2 2 3 3 1 1 4 4 5 5 1	是
2 2 1 1 2 5 6 1 2 2 3 3 1 2 4 3 5 4 5	是 没有
2 4 5 1 2 2 3 3 1 2 4 4 1 5 6 1 2 2 3 1 4 1 5 4 3 5 3	没有 是

## 问题 F. 快速博格索尔特

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
	1024 兆字节

您听说过 Bogosort 吗？Bogosort 是一种非常有趣的随机排序算法，它可以在最佳情况下以  $O(n)$  的时间复杂度对包含  $n$  个元素的数组进行排序。该算法可描述如下：

- 对整个数组进行统一洗牌。
- 检查数组是否排序。

是的，读完上面的算法，你会发现 Bogosort 的预期时间复杂度其实是  $O(n \cdot n!)$ ！

Bogosort 是如此迷人，但运行速度却如此之慢，实在令人遗憾。因此，让我们使用下面的算法 "快速 Bogosort" 对它进行优化，对从 1 到  $n$  的数字排列进行排序，以帮助改进它：

- 如果数组已经排序，则停止。
- 否则，将数组分成尽可能多的段，使得每个段都对应于区间  $[l_i, r_i]$ ，并恰好包含数字  $[l_i, r_i]$  的排列。注意各段不得重叠，它们的结合必须是整个范围  $[1, n]$ 。
- 对于已划分的每个区段  $[l, r]$ ，**如果**  $l < r$ ，则调用 `shuffle(l, r)` 该区段内的数字进行统一洗码。

如果我们只关心 `shuffle(l,r)` 的调用次数，那么很明显，原始 Bogosort 的预期调用次数为  $n!$ 。现在，给定一个从 1 到  $n$  的数字排列请计算 Fast Bogosort 调用 `shuffle(l, r)` 的预期次数。

### 输入

输入的第一行包含一个整数  $n$  ( $1 \leq n \leq 10^5$ )，表示数组的长度。

第二行包含  $n$  个整数  $a_1, a_2, \dots, a_n$ ，表示数组的内容。保证  $a_1, a_2, \dots, a_n$  构成  $1 \sim n$  的排列。

### 输出

输出 Fast Bogosort 调用 `shuffle(l, r)` 的预期次数。可以证明，预期次数总是有理数。此外，在此问题的限制条件下，还可以证明当该值表示为不可约分数  $\frac{p}{q}$  时，我们有  $Q \neq 0 \pmod{q}$

998244353)。因此，存在一个唯一的整数  $R$ ，使得  $R \times Q \equiv P \pmod{998244353}$ ,  $0 \leq R < 998244353$ 。报告这个  $R$ 。

实例

标准输入	标准输出
5 2 1 5 3 4	332748123
10 4 2 3 1 6 5 9 7 10 8	453747445

问题 G. 几何任务

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：1024 兆字节

二维平面上有  $n$  条红线和  $n$  条蓝线。第  $i$  条红线的方程为  $y = a_i x + b_{(i)}$ ，第  $i$  条蓝线的方程为  $x = c_i$ 。

将红线与蓝线配对的值定义为交点的  $y$  坐标。您希望将每条红线与恰好一条蓝线配对，从而得到  $n$  个值。请确定这  $n$  个值的最大可能中值。

长度为  $n$  的数组的中位数是数组中第  $\lceil \frac{n}{2} \rceil$  个最大的元素。例如，数组  $[3, 4, 2]$  的中位数是 3，数组  $[1, 1, 4, 5, 1, ]$  的中位数是 4。

输入

第一行包含测试用例  $T$  的数量 ( $1 \leq T \leq 10^5$ )。测试用例说明如下。每个测试用例的第一行包含一个整数  $n$  ( $1 \leq n \leq 10^5$ )。

每个测试用例的第二行包含  $n$  个整数  $a_1, a_2, \dots, a_n$  ( $-10^9 \leq a_i \leq 10^9$ )。每个测试用例的第三行包含  $n$  个整数  $b_1, b_2, \dots, b_n$  ( $-10^{18} \leq b_i \leq 10^{18}$ )。每个测试用例的第四行包含  $n$  个整数  $c_1, c_2, \dots, c_n$  ( $-10^9 \leq c_i \leq 10^9$ )。保证所有测试用例中  $n$  的总和不超过  $10^5$ 。

输出

对于每个测试用例，输出一个整数 - 可能的最大。

示例

标准输入	标准输出
3 5 0 5 -2 1 2 9 -4 0 10 5 -4 -1 4 -2 4 10 -6 3 1 0 6 -2 -4 3 0 10 22 65 11 1 -34 -1 -39 -28 10 9 1 -2 -5 8 -7 -10 -7 1	9 25 114514

101 48763 651	
---------------------	--



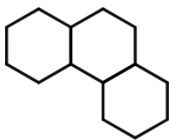
问题 H. 六角拼图

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
	1024 兆字节

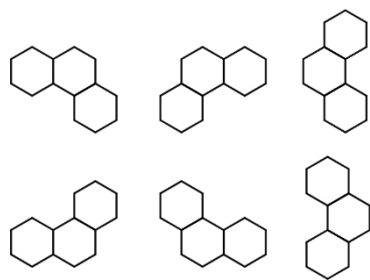
Kotoha 喜欢解谜。，她的朋友 Saki 和 Yui 送给她一个特别的拼图作为生日礼物：一个六边形拼图和许多五颜六色的 V 形拼图块。谜题包含  $n$  行正六边形，其中第  $i$  行正好包含  $i$  个正六边形。每隔  $i > 1$ ，第  $i$  行的第  $j$  个正六边形都会与前一行的第  $j$  个正六边形和  $(j-1)$  第  $j$  个正六边形共享一条边（如果存在的话）。以下是  $n=1, 2, 3, 4$  的谜题示例：



如下图所示，一个 V 形棋子占 3 个六边形。每个棋子的颜色可以是 ABC...XYZ 所代表的 26 种颜色中的一种：



棋子可以旋转 60 度的倍数。更具体地说，V 形棋子有 6 种可能旋转方式，如下图所示：



现在，小叶希望在满足以下条件的情况下，最大限度地增加她所使用的 V 形棋子的数量：

- 她使用的每块 V 形拼图都必须放在拼图内，并且正好占据 3 个六边形。
- 没有两个 V 形棋子占据同一个六边形。
- 相邻的两个 V 形棋子不能使用相同的颜色。如果两个 V 形棋子至少有一条公共边，则这两个

棋子视为相邻。

您能帮她找到 V 形棋子的最大数量和放置方法吗？

## 输入

输入内容只有一行，其中包含一个整数  $n$  ( $1 \leq n \leq 1024$ )，表示谜题的大小。

输出

输出  $n$  行。第  $i$  行包含  $n + i - 1$  个字符。设  $s_{i,j}$  为第  $i$  行中的第  $j$  个字符。您的输出必须符合以下格式：

- 如果  $j \leq n - i$ ，则  $s_{i,j}$  为空格（ASCII 码 32）。
- 如果  $j - (n - i)$  是偶数整数，则  $s_{i,j}$  是一个空间。
- 如果第  $i$  行的第  $(\frac{j-(n-i)+1}{2})$ - 个六边形没有被任何 V 形棋子覆盖，则  $s_{i,j}$  是一个点 ASCII 码：“.”（代码 46）。
- 否则， $s_{i,j}$  用来表示覆盖  $(\frac{j-(n-i)+1}{2})$ -th 的 V 形片的颜色。  
第  $i$  行中的六边形，它将是一个大写英文字母。如果满足以下所有条件，

2

您的结果将被视为正确：

- 它符合上述所有格式。
- 您放置的所有棋子都是 V 形的。
- 相邻两个 V 形棋子的颜色不能相同。
- 您放置的棋子数量是上限。

请勿在每行末尾输出额外的空格，否则您的解决方案可能会被视为错误。

实例

标准输入	标准输出
2	. ..
3	W W. .W.
4	. .R .B R B B R .

问题 1. 寻找终极艺术品

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 1024 兆字节

贾斯汀是一位技艺精湛的法师，他花了数年时间收集了  $n$  件神秘的工艺品。每件神器都被赋予了一定的力量，用一个非负整数表示。最近，贾斯汀偶然发现了一种名为 "K 融合 "的传奇法术，据说这种技术能将他的神器提升到前所未有的水平。

使用 "k-融合 "法术，贾斯汀可以将  $k$  件神器融合成一件神器。这个新形成的神器的力量是被消耗的  $k$  件力量的乘积。不过，这个过程是不可逆的，因为原来的  $k$  件神器会在融合过程中永远消失。

贾斯汀决心用这个咒语制作出最强大的神器。他执行了一定数量（可能为零）的 k 融合，从而最大限度地提高了剩余神器的最大威力。完成后，贾斯汀向你挑战，计算剩余工艺品的最大威力。

但有一个转折！结果可能是一个天文数字，而贾斯汀不想处理如此巨大的数值。相反，他要求你计算除以 998244353 的最大幂的余数--这个质数在他的魔法研究中意义重大。

你能帮助贾斯汀确定在他的 k 融合序列之后，剩余神器的最大可能功率的剩余部分吗？

输入

该问题包含多个测试用例。第一行输入包含一个整数  $T$  ( $1 \leq T \leq 1000$ )，表示测试用例的数量。

每个测试用例都有两行输入。第一行输入包含两个整数  $n, k$  ( $2 \leq n \leq 2 \times 10^5, 2 \leq k \leq n$ )，表示初始的人工制品数量和 k 融合参数。

第二行输入包含  $n$  个非负整数  $p_1, p_2, \dots, p_n$  ( $0 \leq p_i \leq 10^9$ )，其中  $p_i$  是第  $i$  个神器的幂。

保证所有测试用例的  $n$  之和不超过  $2 \times 10^5$ 。

输出

对于每个测试用例，在一行中输出一个整数，表示贾斯汀挑战的答案。

示例

标准输入	标准输出
3	923923948
8 3	100
44 5 2018 8 8 2024 8 28	0
5 4	
4 5 5 1 0	
5 2	

00000	
-------	--

## 问题 J. 及时渲染分析

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
	1024 兆字节

网站包含各种元素，包括可见的用户界面元素和不可见的分组框架。正如您在 DOMjudge 团队的界面上看到的那样，它可以分为多个组件。这些区域以分层树形结构（称为 DOM 树）组织起来，尽管它们之间并无直接关联。

您是一个全新浏览器项目的开发人员。在该项目中，元素被建模为放置在笛卡尔平面上的矩形，并以适当的方式嵌套。也就是说，每对矩形要么相互严格包含，要么完全不相交。

目前的算法以迭代的方式运行：浏览器在一次迭代中渲染最外层的元素，这些元素不包含任何其他元素。这些元素会被标记为已完成，并在随后的迭代中被忽略。这一过程不断重复，直到所有元素都渲染完毕。因此，浏览器计算一个元素的 *呈现深度* 时，会比严格 *包含* 该元素的所有其他元素的最大呈现深度多一倍，如果没有这样的元素，则该元素的呈现深度为零。

然而，现代网站通常具有更多的动画，使当前的渲染算法不堪重负。为了找出性能瓶颈，您的团队需要一种工具来分析网站结构。每个元素都有一个动画开关，用于控制是否显示动画。为了更好地查看页面，元素的一个关键属性就是它的任何部分是否显示动画。因此，如果一个元素包含某些动画开关处于打开状态的元素（包括它本身），那么这个元素就被称为动画元素。

为了衡量渲染过程的迭代速度可能有多慢，我们赋予了您监控每次迭代的资源使用情况的任务。更具体地说，您需要处理动画开关的切换，并回答询问给定渲染深度下 *动画* 元素数量的查询。

### 输入

第一行输入包含两个整数  $n$  ( $1 \leq n \leq 5 \times 10^5$ ) 和  $q$  ( $1 \leq q \leq 5 \times 10^5$ )，其中  $n$  是元素个数， $q$  是即将发生事件的总数。

接下来的  $n$  行中，每一行都包含四个整数  $x_{(1)}$ 、 $y_{(1)}$ 、 $x_{(2)}$ 、 $y_{(2)}$  ( $1 \leq x_{(1)} < x_{(2)} \leq 10^9$ ,  $1 \leq y_{(1)} < y_{(2)} \leq 10^9$ )，表示元素是一个左下角  $(x_{(1)}, y_{(1)})$  和右上角  $(x_{(2)}, y_{(2)})$  的矩形。矩形的边界永远不会相交，甚至不会相交于一点。

之后是  $q$  行，表示按时间顺序排列的事件。第一种格式表示应按输入顺序切换第  $i$  个元素 ( $1 \leq i \leq n$ ) 的动画开关，即打开或关闭，反之亦然。第二种格式表示查询渲染深度为  $k$  的动画元素数量 ( $0 \leq k$ )。

初始情况下，所有动画开关都处于关闭状态，并且至少有一个第二种格式的事件。查询中指定的

呈现深度不会超过所有元素的最大呈现深度。

## 输出

对于每个查询，用一行输出具有给定渲染深度的动画元素的数量。

示例

标准输入	标准输出
8 7	1
1 1 15 11	2
2 2 7 4	3
2 5 7 10	1
3 6 4 9	
5 6 6 7	
8 2 14 10	
9 5 13 9	
11 6 12 7	
^4	
^5	
^8	
?0	
?1	
?2	
?3	



## 问题 K. 夜骑士

输入文件：	标准输入
输出文件：	标准输出
时间限制	7 秒 内存限制
	1024 兆字节

在幻境的第 16 个夜晚，骑士们聚集一起，准备一决雌雄。

有两组骑士准备战斗，每组由  $n$  名骑士组成。第一组中第  $i$  个骑士的力量为  $a_i$ ，第二组中第  $j$  个骑士的力量为  $b_j$ 。

您的任务是组织今晚骑士之间的战斗。每场格斗由每组的一名骑士参加，每名骑士最多只能参加一场格斗。格斗的结果并不重要，真正重要的是格斗变得多么疯狂。

格斗的 lunaticus 定义为参与格斗的两名骑士的力量总和。不过，有一个特殊条件：如果战斗的 lunaticus 达到 998244353，它就会溢出并重置为零。具体来说，第一组第  $i$  位骑士和第二组第  $j$  位骑士之间的战斗的 lunaticus 为  $(a_i + b_j) \bmod 998244353$ 。

该特殊条件仅适用于单场比赛的幸运值。在计算多场战斗的幸运值总和时，幸运值总和可超过 998244353。

由于比赛场次尚未确定，您想计算一下，对于 1 到  $k$  每个整数  $x$ ，今晚正好举行  $x$  场比赛的最大可能的 lunaticus 总和是多少。，还有  $m$  对特定比赛无法组织。请在避开这  $m$  个限制条件的前提下，最大限度地计算 lunaticus 的总和。

### 输入

第一行包含三个整数  $n$ 、 $m$ 、 $k$  ( $1 \leq n \leq 10^5$ ,  $0 \leq m \leq 3 \times 10^5$ ,  $1 \leq k \leq \min(n, 200)$ )，分别代表每组的骑士人数、无法组织的额外战斗对数以及需要回答的问题数。

第二行包含  $n$  个整数  $a_1, a_2, \dots, a_n$  ( $0 \leq a_i < 998244353$ )，代表第一组中骑士的实力。

第三行包含  $n$  个整数  $b_1, b_2, \dots, b_n$  ( $0 \leq b_i < 998244353$ )，代表第二组中骑士的实力。

接着是  $m$  行，其中第  $i$  行包含两个整数  $u_i, v_i$  ( $1 \leq u_i, v_i \leq n$ )，表示不能由第一组中的第  $u_i$  个骑士和第二组中的第  $v_i$  个骑士这一对组织战斗。保证所有给定的配对都是不同的。

### 输出

在一行中输出  $k$  个数字，其中第  $i$  个数字代表今晚正好举行  $i$  场格斗的最大可能的 lunaticus 总和。对于第  $i$  个数字，如果不可能组织  $i$  场比赛，则输出 -1。

实例

标准输入	标准输出
3 4 3 10 998244352 5 998244352 8 6 1 2 1 3 2 2 3 3	998244351 998244364 27
6 10 5 749208241 448597025 773867529 808779198 281439035 850615248 796547348 327547103 1 4 5 3 3 4 2 1 5 6 4 3 6 2 6 3 2 6 2 4	882168541 1667618296 2328768389 2900938 554621438 19807774 765641981 702406342
1 1 1 0 0 1 1	-1

13 3354309143

## 问题 L. 懒人苏珊

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
	1024 兆字节

懒人餐盘是放在餐桌上的一个可旋转的圆形托盘，用来盛放食物或餐具，方便围坐在餐桌旁的每个人分享和盛放自己的食物。

有  $n$  个人围坐在一张圆桌旁享用晚餐，每个人的顺时针编号从 0 到  $n-1$ ， $n$  个人之间的距离相同。请注意，顺时针方向第  $(n-1)$  个人旁边的人是第 0。

晚餐由  $n$  道菜组成，每道菜的编号从 0 到  $n-1$ ，按上菜顺序排列。服务员将从第  $x$  个人开始，按顺时针顺序把每道菜端到每个人面前的懒人沙发上。具体来说，第  $i$  道菜将放在  $[(x+i) \bmod n]$ -th 的人面前。

每次旋转懒人沙发时，所有菜肴都可以顺时针或逆时针旋转到下一个人的位置。更确切地说，原本在第  $i$  个人前面的盘子在顺时针旋转一次后会移动到第  $[(i+1) \bmod n]$ - 个人的位置，而在逆时针旋转一次后会移动到第  $[(i+n-1) \bmod n]$ - 个人的位置。每次操作耗时 1 秒。

每个人都有一个可达距离  $r_i$ 。第  $i$  个人可以享用当前在他面前的菜肴如果存在一个满足以下条件的整数  $k$ ，则第  $j$  人将是第  $j$  人：

- $-r_i \leq k \leq r_i$
- $(i+k+n) \bmod n = j$

满足上述条件后，人们就可以立即享用这道菜了。

$n$  有  $m$  个偏好。第  $i$  个偏好是第  $p_i$ - 人希望在所有菜肴上桌后  $t_i$  秒之前享用第  $d_i$ - 道菜。如果服务员从第  $x$  人开始上菜，且所有  $x$  在 0 到  $n-1$  之间，请判断是否能满足所有偏好。

### 输入

第一行包含测试用例  $T$  的数量 ( $1 \leq T \leq 2500$ )。测试用例说明如下。

每个测试用例的第一行包含两个整数  $n$ 、 $m$  ( $2 \leq n \leq 5000$ ,  $0 \leq m \leq \min(n^2, 10^5)$ )，表示人数和偏好数。

每个测试用例的第二行包含  $n$  个非负整数  $r_0, r_1, \dots, r_{n-1}$  ( $0 \leq r_i \leq n$ )，表示第  $i$  个人的到达距离。

每个测试用例的后  $m$  行包含三个整数  $p_i, d_i, t_i$  ( $0 \leq p_i < n$ ,  $0 \leq d_i < n$ ,  $1 \leq t_i \leq 10^9$ )，表示第  $i$  个偏好的信息。保证每  $1 \leq i < j \leq m$ ,  $p_i \neq p_j$  或  $d_i \neq d_j$ 。

保证所有测试用例的  $n$  之和不超过 5000。保证所有测试用例的  $m$  之和不超过  $10^5$ 。

## 输出

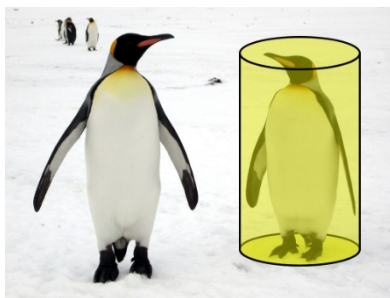
对于每个测试案例，输出长度为  $n$  的字符串  $s$ ，  
 $s_x = 1$ 。否则  $s_x = 0$ 。

示例

标准输入	标准输出
4	10100
5 6	11111010
0 0 1    2 1	0000
3 2 2	111111111111
0 0 2	
4 4 4	
1 1 3	
4 3 5	
0 3 2	
8 10	
2 2 2    0 3 0 0 1	
4 3 1	
1 0 1	
0 7 6	
4 4 1	
1 7 5	
0 4 2	
5 3 4	
4 6 1	
7 7 3	
0 2 4	
4 4	
0 0 0    0	
1 0 2	
2 0 2	
0 0 2	
3 0 2	
13 0	
1 1 4    5 1 4 1 9 1 9 8 1 0	

## 问题 M. 企鹅的机器学习

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



近似企鹅，原图由 David Stanley 提供，根据 CC BY 2.0 许可使用。

企鹅是一种不会飞的海洋鸟类，几乎只生活在南半球。这些游泳健将一直是动物园里最受欢迎的物种之一，但不幸的是，这种可爱的生物正受到气候变化的威胁。因此，国际企鹅保护中心（ICPC）发起了一个研究如何拯救企鹅的项目。他们启动了最先进的自动驾驶车辆--大量的漫游车（带伪装），在其基地周围收集图像，用于研究。通过将不同视角的多张图片结合起来，可以在三维模型下更系统地研究这些鸟类。然而，缺点是这种计算显然无法通过人力完成，这就是机器学习的用武之地。不幸的是，它并没有成功。机器在做标记的时候，完全是在胡编乱造。

也许是模式出了问题。然而，ICPC 需要彻底改革其培训方法。他们提出了许多建议，其中大部分都很容易实现。然而，他们发现有一个问题特别具有挑战性：他们希望对模型标记出不像企鹅的形状进行惩罚，但究竟是什么让一个形状看起来像企鹅呢？

原来，物理学家几年就知道了：这是一个圆柱体！

要检查是否有一个圆柱面经过算法标记的所有点，这有多难呢？事实是，几个月来一直没有有效的结果。因此，他们降低了短期目标，决定只输入企鹅笔直站立图片。也就是说，圆柱体仅限于直圆柱体（圆柱面与圆底垂直的圆柱体），其底面位于  $x$ - $y$  平面上。当然，企鹅既不是虚数，也不是无限大。应排除体积为零或无限大的圆柱体。知名计算机科学家和几何大师，你们能解决这个问题吗？

### 输入

第一行包含一个数字  $n$  ( $1 \leq n \leq 10^5$ )，即模型标记的点数。下面  $n$  行中的每一行都包含三个整数  $x$ 、 $y$  和  $z$  ( $-10^9 \leq x, y \leq 10^9, 0 \leq z \leq 10^9$ )，表示三维空间中有一个坐标为  $(x, y, z)$  的点。输

入中不会有重复的点。

## 输出

如果有一个圆柱体满足所有限制条件，则在第一行输出 "可能"。否则，在一行中输出 "不是企鹅"。

实例

标准输入	标准输出
5 0 0 0 3 4 1 -3 -4 2 -3 4 3 3 -4 4	也许
4 0 6 2 3 4 1 -3 -4 1 -3 4 1	不是企鹅
2 0 0 0 1 1 0	也许