

Problem A. 无线网络整点栅格统计

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

在无线网络的规划或仿真中，最常见的手段是栅格化，即将地图放置于平面直角坐标系上，将不同区域划分到不同的栅格内。其中一种栅格化的方式需要借助整点。所谓整点，指的是一个坐标为 (x, y) 的点，当且仅当 $x \in \mathbb{Z}, y \in \mathbb{Z}$ ，其中 \mathbb{Z} 表示整数集合。如果平面上一个正方形的四个顶点都是整点，那么我们称这个正方形是整点栅格。

现考虑一块 $n \times m$ 的规划区域，该区域可表示为 $\{(x, y) | 0 \leq x \leq n, 0 \leq y \leq m\}$ 。网规工程师时不时会给出该区域内的一个整点的坐标 $(a, b), 0 \leq a \leq n, 0 \leq b \leq m$ ，并询问你，以 (a, b) 为一个顶点，能在该区域规划出多少个不同的面积为正的整点栅格？

由于规划过程中需要反复尝试，询问次数实在太多，你决定直接告诉他所有 $(a, b), 0 \leq a \leq n, 0 \leq b \leq m$ 询问的答案。请输出一个 $(n+1) \times (m+1)$ 的矩阵作为你的答案。

Input

输入包含 1 行，两个整数 $n, m (1 \leq n, m \leq 100)$ ，表示规划区域的大小。

Output

输出包含 $n+1$ 行，每行包含 $m+1$ 个整数，每行的整数之间用空格隔开。

其中第 $i (1 \leq i \leq n+1)$ 行第 $j (1 \leq j \leq m+1)$ 个整数表示以 $(i-1, j-1)$ 为一个顶点，能在规划区域内规划出的不同整点栅格的数量。

Examples

standard input	standard output
1 1	1 1 1 1
2 2	2 3 2 3 4 3 2 3 2
4 5	4 8 10 10 8 4 7 11 13 13 11 7 8 12 14 14 12 8 7 11 13 13 11 7 4 8 10 10 8 4

Note

注意到，根据题意，整点栅格的边并不一定平行于坐标轴。

样例 $n=2, m=2$ 中，考虑以 $(0, 1)$ 作为顶点，则下面存在 3 个合法的整点栅格：

- 以 $(0, 1), (1, 1), (1, 0), (0, 0)$ 为顶点的整点栅格。
- 以 $(0, 1), (0, 2), (1, 2), (1, 1)$ 为顶点的整点栅格。
- 以 $(0, 1), (1, 2), (2, 1), (1, 0)$ 为顶点的整点栅格。

因此，该样例答案矩阵的第 1 行第 2 个整数为 3。

Problem B. 异或和之和

Input file: standard input
Output file: standard output
Time limit: 8 seconds
Memory limit: 1024 megabytes

给定一棵 n 个点的树，点有点权，记第 i 个点的点权为 w_i 这棵树的权值为：

$$\sum_{i=1}^n \sum_{j=i}^n PathXorsum(i, j)$$

其中 $PathXorsum(u, v)$ 表示 u 到 v 这条链上所有点的点权 w_i 的按位异或和。

再有 q 次操作，每次操作给定 x, v ，把 x 节点的点权 w_x 改成 v 。

对于第一次操作前以及每一次操作之后，都求一下这棵树当前的权值。

Input

第一行一个整数 n ($1 \leq n \leq 10^5$)，表示树的大小。

第二行 n 个整数 w_1, w_2, \dots, w_n ($0 \leq w_i \leq 10^8$)，表示每个点初始的点权。

接下来 $n - 1$ 行，每行两个整数 u, v ($1 \leq u, v \leq n$)，表示树上一条从 u 到 v 的边。

接下来一行一个整数 q ($0 \leq q \leq 10^4$)，表示操作个数。

接下来 q 行，每行两个整数 x, v ($1 \leq x \leq n, 0 \leq v \leq 10^8$)，表示一次把 x 节点的权值 w_x 改成 v 的操作。

Output

输出共 $q + 1$ 行，每行一个整数，表示给定的树当前的权值。

Example

standard input	standard output
4	32
1 2 3 4	32
1 2	34
1 3	
2 4	
2	
1 0	
4 7	

Problem C. 无线基站最佳选址

Input file: standard input
Output file: standard output
Time limit: 4 seconds
Memory limit: 256 megabytes

华为的无线基站为了做到更好的用户覆盖和更低的建站开销，往往需要对基站的站点选址提前做好规划，在满足所有用户需求的同时，要尽可能减少建立基站的花费。

现在某块区域，作为基站项目负责人的你，需要建立两个信号基站来覆盖该区域的所有用户。这两个基站的信号辐射方式是不同的，基站A的信号辐射区域是一个圆形区域，基站B的信号辐射区域则是一个正方形区域。而建立两个基站的花费和它们辐射区域的面积是正相关的。

而为了保证该目标区域的所有用户都能被信号覆盖，所有用户都必须位于两个基站中的其中一个基站的辐射区域内（包含边界），在达成该目标的同时，也要使得两个基站的辐射区域面积之和尽可能的小，从而降低建站的费用。

Input

第一行一个整数 n ($1 \leq n \leq 80$)，表示该区域内用户的个数。

接下来 n 行，每行两个整数 x, y ($|x|, |y| \leq 10000$)，表示每个用户的坐标 (x, y) 。

保证没有两个用户位于同一个位置。

Output

输出一个实数，代表两个基站的辐射区域面积之和的最小值。

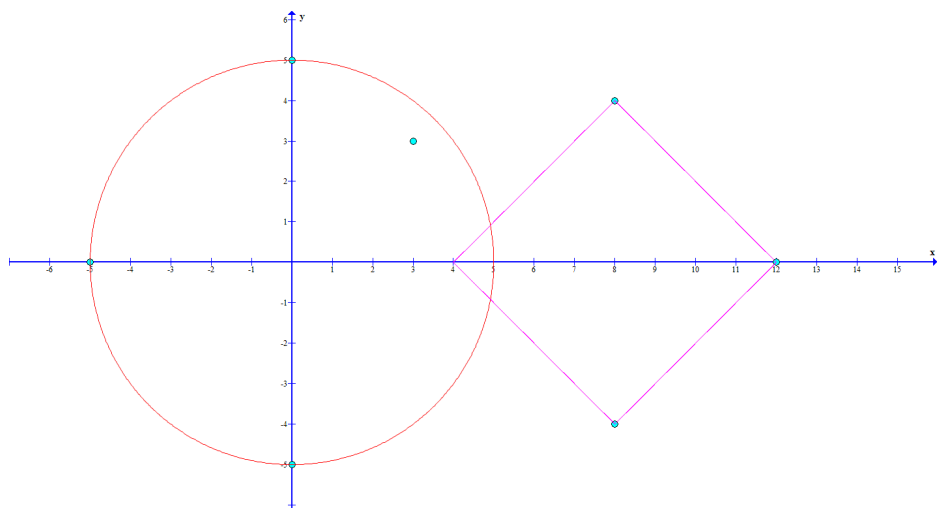
记 a 为选手输出， b 为标准输出，如果满足 $\frac{|a-b|}{\max(b, 1)} \leq 10^{-6}$ 则认为答案正确。

Example

standard input	standard output
7 0 5 -5 0 0 -5 3 3 8 4 8 -4 12 0	110.539816340

Note

可以建一个圆心为 $(0, 0)$ ，半径为 5 的圆基站，以及顶点为 $(8, 4), (12, 0), (8, -4), (4, 0)$ 的正方形基站，总面积为 $25\pi + 32$ 。



Problem D. 咸鱼跑酷

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

有一个长为 n 的跑道，位置 i 有两个道具 $a_{i,0}, a_{i,1}$ ，其中每个道具是 $+x$ ， $*x$ 这两种中的一种。
现给定 q 组询问，每次给定三个数字 u, l, r ，表示 u 个人要从跑道的位置 l 开始跑酷到位置 r ，每到达一个位置（包括起点和终点），便可选取该位置上两个道具中的一个，并以此改变当前的人数，问从 l 走到 r 人数最多可能可以变成多少，输出模 998244353 的结果。

Input

第一行一个整数 n ($1 \leq n \leq 10^5$)，表示跑道的长度。
接下来 n 行，每行两个字符串 $a_{i,0}, a_{i,1}$ ，每个字符串由首位运算字符 op ($op \in \{+, *\}$) 和后续数字 x ($1 \leq x \leq 10^9$) 组成，表示位置 i 上的道具。
接下来一行一个整数 q ($1 \leq q \leq 10^5$)，表示询问组数。
接下来 q 行，每行三个整数 u, l, r ($1 \leq u \leq 10^9, 1 \leq l \leq r \leq n$)，表示一次询问。

Output

输出共 q 行，每行一个整数，表示询问答案模 998244353 的结果。

Example

standard input	standard output
5	144
+3 +1	16
+4 *1	
*5 *9	
*2 +6	
+5 +3	
2	
1 1 4	
5 4 5	

Note

- 对于第一组询问，可以分别选取 $+3, +4, *9, *2$ 四个道具，最终人数为 $((1 + 3) + 4) \times 9 \times 2 = 144$ 。
- 对于第二组询问，可以分别选取 $+6, +5$ 两个道具，最终人数为 $(5 + 6) + 5 = 16$ 。

特别注意： 根据输入格式，可以知道数据中可能存在 $x = 998244353$ 或 $u = 998244353$ 的情况。

Problem E. 无线软件日

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

华为上海研究所内每个月都有各种各样的小活动。这不，一年一度的无线软件日来了。
今年的软件日有一份大礼：HUAWEI mate70 非凡大师。所有员工都可以通过抽奖的方式来抽取大奖，但是参与者总是要面临各种困难的，比如今年主办方就别出心裁的从某开源代码中节选下来了一段文字。你可以通过一把剪刀把文字中的任意字母剪下来再重新组合。
每当你能够组合出来一个“Shanghai”（不区分大小写），即可获得一个奖券。请问给出对应的节选文字后，你最多能够获得多少张奖券呢？

Input

第一行一个整数 n ($1 \leq n \leq 10^6$)，表示文字段中字母个数。
第二行一个由大写字母和小写字母组成的字符串 S ，表示文字段中所有字母。

Output

输出一行一个整数，表示最多获得多少张奖券。即最多可以拼出多少个“Shanghai”（大小写不敏感，比如拼成“shAnGHaI”也可以）。

Example

standard input	standard output
27 ShangHaiShiSaiHaiGeTongKuai	2

Note

- 第一张可以凑成“ShangHai”，见下图大字加粗部分。
- 第二张可以凑成“ShanGHai”，见下图大字斜体部分。

Shang**Hai** *Shi**sai* *Hai*GeTo*ng*Kuai

Problem F. 羁绊大师

Input file: standard input
Output file: standard output
Time limit: 3 seconds
Memory limit: 256 megabytes

在某一自走棋游戏中，胖胖龙此刻拥有 n 个英雄，其中第 i 个英雄拥有两种羁绊，分别为 $a_i, b_i (a_i < b_i)$ 。不存在两个英雄拥有完全相同的两种羁绊，即对于 $1 \leq i < j \leq n$ ，有 $a_i \neq a_j$ 或 $b_i \neq b_j$ 。因为该游戏模式的特殊性，一共具有 m 种羁绊，且每种羁绊至多只有两个英雄拥有。当胖胖龙处于等级 L 时，可以选择自己的 L 个英雄上阵。对于每种羁绊，当且仅当上阵英雄中有两个英雄拥有此羁绊时，此羁绊为激活状态。

胖胖龙的梦想是成为羁绊大师，他希望选出 L 个英雄上阵，使得自己激活的羁绊种数尽可能多。胖胖龙希望你能帮帮他，如果你帮助了他，他会给你跳一段节奏鲜明而富有动感的舞蹈。

请你分别对 $L \in \{1, 2 \dots n\}$ 的 n 种情况，分别输出激活羁绊的最大数量。

Input

第一行，包含两个非负整数 $n, m (1 \leq n \leq 10^5, n \leq m \leq 2n)$ 分别表示胖胖龙此时拥有的英雄数量，以及游戏中存在羁绊的数量。

接下来 n 行，第 i 行包含两个整数 $a_i, b_i (1 \leq a_i < b_i \leq m)$ 分别表示第 i 个英雄拥有的两种羁绊。

Output

输出一行，包含 n 个整数，从左到右第 i 个整数表示当 $L = i$ 时，胖胖龙最多能够激活的羁绊种数。整数之间用一个空格隔开。

Examples

standard input	standard output
3 6 1 2 2 5 1 6	0 1 2
10 10 1 2 2 3 1 3 4 5 5 6 6 7 4 7 8 9 9 10 8 10	0 1 3 4 4 6 7 7 8 10

Note

样例 1 中：

- 当 $L = 1$ 时，无法激活任何羁绊。
- 当 $L = 2$ 时，上阵英雄 1 和英雄 2，则可以激活 2 号羁绊。
- 当 $L = 3$ 时，所有英雄上阵，最多能同时激活 1 号羁绊和 2 号羁绊。

Problem G. 象棋大师

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

小万和小宁在玩一种很新的象棋。

棋盘的大小是 $n \times n$ ，棋子仅能在 $\{(a, b) | 0 \leq a, b \leq n\}$ 范围的整点上移动。

小万只有一只卒，位于起点 $(0, 0)$ ，这只卒每次移动可以向 x 轴或 y 轴增大的方向移动。换言之，假设卒此刻位于 (x, y) 那么他下一步只能移动到 $(x + 1, y)$ 或 $(x, y + 1)$ ，另外，移动到棋盘外是不允许的。

小宁有 m 只马，这些马的位置已知，马不能移动，但是一旦小万的卒进入到了小宁任何一只马的攻击范围时，小宁的卒会立即被吃掉。马的攻击范围与中国象棋规则相同，包含了“别马脚”的情况。形式化来说，假如一只马位于 (u, v) ，那么：

- 在 $(u + 1, v)$ 位置没有其它棋子时，该马可以攻击到 $(u + 2, v - 1)$ 和 $(u + 2, v + 1)$
- 在 $(u - 1, v)$ 位置没有其它棋子时，该马可以攻击到 $(u - 2, v - 1)$ 和 $(u - 2, v + 1)$
- 在 $(u, v + 1)$ 位置没有其它棋子时，该马可以攻击到 $(u - 1, v + 2)$ 和 $(u + 1, v + 2)$
- 在 $(u, v - 1)$ 位置没有其它棋子时，该马可以攻击到 $(u - 1, v - 2)$ 和 $(u + 1, v - 2)$

现在，小万希望将位于起点 $(0, 0)$ 的卒，安全的移动到终点 (n, n) ，在这个过程中（包括起点和终点）不被任何一只马攻击到，请问他有多少种不同的走法？两个走法不同，当且仅当存在 $1 \leq i \leq 2n$ 满足两个走法的第 i 步的移动方向不同。

注意到，小万可以在移动卒的过程中吃掉小宁的马。只要卒移动到小宁的马所在位置，这只马就会被吃掉。被吃掉的马不再能够攻击，并从棋盘上消失。

Input

第一行包含两个正整数 n, m ($1 \leq n \leq 100, 1 \leq m \leq 10$)，分别表示棋盘大小和小宁马的数量。

接下来 m 行，第 i 行包含两个整数 x_i, y_i ($0 \leq x_i, y_i \leq n$)，表示第 i 只马的位置。保证马的位置不相互重合，且不与卒的起点 $(0, 0)$ 重合。

Output

输出一行答案，一个整数，表示卒从起点安全走到终点的方案数。答案对 998244353 取模。

Examples

standard input	standard output
2 2 1 1 1 2	6
2 2 1 0 0 1	0
2 8 0 1 0 2 1 0 1 1 1 2 2 0 2 1 2 2	2

Note

样例 1 解释，因为 $(1,1)$ 的马别住了 $(1,2)$ 的的马脚，因此存在以下合法方案：

- $(0,0) \rightarrow (1,0) \rightarrow (2,0) \rightarrow (2,1) \rightarrow (2,2)$
- $(0,0) \rightarrow (1,0) \rightarrow (1,1) \rightarrow (2,1) \rightarrow (2,2)$
- $(0,0) \rightarrow (1,0) \rightarrow (1,1) \rightarrow (1,2) \rightarrow (2,2)$
- $(0,0) \rightarrow (0,1) \rightarrow (1,1) \rightarrow (2,1) \rightarrow (2,2)$
- $(0,0) \rightarrow (0,1) \rightarrow (1,1) \rightarrow (1,2) \rightarrow (2,2)$
- $(0,0) \rightarrow (0,1) \rightarrow (0,2) \rightarrow (1,2) \rightarrow (2,2)$

共 6 种合法方案。

Problem H. 出金记录

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

有一个卡池，可以抽出两种卡——白卡和金卡。卡池有一个计数器 n ，每次抽到金卡后会清零，而抽到白卡后值会加 1。当计数器的值为 n 时，下一次抽到金卡的概率为 p_{n+1} ，抽到白卡的概率则为 $1 - p_{n+1}$ 。计数器的初始值为 0。

$p_i (i \geq 1)$ 的计算方式如下：

$$p_i = \begin{cases} p_0 & 1 \leq i \leq B \\ \min\{1, p_0 + (i - B)v\} & i \geq B + 1 \end{cases}$$

其中 p_0, v 为有理数。 B 为已知整数。

此外，卡池会为每个玩家提供一个出金记录功能，记录玩家最近 k 次出金的所用抽数。

但是记录空间有限，卡池会仅会维护最近 k 次的出金记录，用 k 个整数 a_1, a_2, \dots, a_k 表示。其中第 i 个整数表示最近第 i 次抽到金卡所用的抽数（即最近第 i 次出金距离上一次出金后到该次出金后所用的抽数）。特别的，如果最近第 i 次出金是第一次出金，则表示从开始抽卡到第一次出金后所用的抽数）。

特别的，如果一个玩家出金次数 g 不足 k 次，则他的记录会用 0 来填充 $a_j (g < j \leq k)$ 。

你是一个新玩家，还没有进行过抽卡。你在抽卡之前看到了一份其他玩家的出金记录 $a_1, a_2 \dots a_k$ ，你知道期望抽出多少个金后能（首次）让自己的出金记录和他的一模一样？

Input

第一行，一个整数 $B (1 \leq B \leq 10^5)$ ，意义如题目描述所示。

第二行，四个整数 $a, b, c, d (0 \leq a \leq b < 998244353, 1 \leq b, 0 \leq c \leq d < 998244353, 1 \leq d)$ ，用来表示 p_0 和 v ，具体的， $p_0 = \frac{a}{b}, v = \frac{c}{d}$ 。

第三行，一个整数 $k (1 \leq k \leq 10^5)$ ，表示出金记录的长度。

下面 k 行，第 i 行有一个整数 $a_i (1 \leq a_i \leq 10^5)$ ，表示倒数第 i 次出金所用的抽数。

Output

若答案为有理数，输出一个整数，表示它对 998244353 取模的值。具体来说，若答案为 $\frac{p}{q}$ ，请输出整数 x 以满足 $0 \leq x < 998244353$ 且 $qx \equiv p \pmod{998244353}$ 。

否则，输出 -1。

Examples

standard input	standard output
1 1 2 0 1 2 1 2	8
1 1 2 1 2 2 1 1	6
73 6 1000 6 100 1 91	-1

Problem I. 六元组计数

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

给定一个质数 P 和一个正整数 N ，问有多少个不同的六元组 (a, b, c, p, q, r) 满足：

- 1. $a, b, c, p, q, r \in \mathbb{Z}$
- 2. $0 \leq a, b, c < P$ 且 $0 < p, q, r \leq N$
- 3. $a^p + b^q \equiv c^r \pmod{P}$

输出答案对 998244353 取模的结果。

Input

一行，两个正整数 P, N 。
 $2 \leq P \leq 10^5$ 且 P 为质数。
 $1 \leq N \leq 10^9$ 。

Output

一行，一个正整数，表示满足条件的六元组的个数对 998244353 取模的结果。

Examples

standard input	standard output
3 3	243
17 19	1982891
23 114514	323464244

Problem J. 极简合数序列

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

在未来的无线通信技术领域，一种革命性的信号加密技术——"合数核心加密"正在研发之中。这项技术的核心在于利用连续的合数序列作为信号加密的关键元素，以增强数据传输的安全性。为了实现高效且资源节约型的加密模块，我们需要设计一套算法，用于从接收到的一系列正整数信号中，提取出能构成最短连续合数序列的区域，我们称之为"合数核心"。

作为无线算法专家，你的任务是编写一个算法，分析接收到的正整数数组 a_1, a_2, \dots, a_n ，识别出能够构成最小尺寸合数核心的子段。具体来说，你需要找到一个区间 $[l, r]$ （其中 $1 \leq l \leq r \leq n$ ），使得从 a_l 到 a_r 的区间所有整数之和为一个合数，并且此区间两端的距离 $r - l$ 达到最小，请你输出核心的最小尺寸（即 $r - l$ 的最小值）。

Input

该题目每个测试点包含多组数据，第一行首先输入一个正整数 T 表示数据组数。

对于每组数据：

- 第一行输入正整数 $n(1 \leq n \leq 1000)$ ，表示数组长度。
- 第二行，包含 n 个正整数，表示 a_1, a_2, \dots, a_n ，满足 $1 \leq a_i \leq 1000, (1 \leq i \leq n)$ 。

数据保证，所有数据中 n 的总和不超过 1000，即 $\sum n \leq 1000$ 。

Output

输出 T 行，第 i 行表示第 i 组数据的答案，即能找到的合数核心的最小尺寸。特殊的，如果不存在合数核心，请输出 -1 。

Example

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

Note

样例解释：

- 第一个序列，只有一个区间 $[1, 1]$ ，元素和为 4，是合数区间，因此输出 $1 - 1 = 0$ 。
- 第二个序列，只有一个区间 $[1, 1]$ ，元素和为 5，不是合数区间，因此输出 -1 。
- 第三个序列，最小的合数核心为区间 $[1, 3]$ ，元素和为 6，是合数，因此输出 $3 - 1 = 2$ ，可以发现不存在更小的合数核心。

Problem K. 时光

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

一个人生来带着 N 个使命。
当他选择面对第 i 个使命时，则需要消耗他 A_i 的时光，并在此期间不间断地完成这个使命。
每当他完成一个使命时，这一时刻，他会瞬间想起曾经完成过的所有使命，并从中获得回忆。具体来说，当他在选择面对一个使命并完成后，如果曾经他完成过第 i 个使命，则此刻他将获得 B_i 的回忆。
每个使命只能被面对并完成至多一次。他也可以选择面对使命，度过任意长的时光。他的人生旅途共有长度为 M 的时光。在旅途结束之前，他希望获得的回忆的总和尽可能多。

Input

输入第一行包含两个整数 N, M ($1 \leq N \leq 30, 1 \leq M \leq 10^9$)，分别表示使命的数量和时光的长度。
第二行包含 N 个整数 A_i ($1 \leq A_i \leq 10^9$)，表示完成每个使命所需的时光。
第三行包含 N 个整数 B_i ($1 \leq B_i \leq 10^9$)，表示完成过的使命能在未来每个面对使命的时刻所能带来的回忆。

Output

输出仅包含一个整数，表示最大可能的总回忆。

Examples

standard input	standard output
5 20 4 5 6 7 8 5 8 7 4 9	26
1 100 101 101	0

Note

提示： 由题意可知，当完成一个使命时，该使命不会立即产生回忆。对于样例1，可以先选择第5个使命，完成获得0回忆，再选择第2个使命，完成获得9回忆，再选择第1个使命，完成后获得9+8=17回忆，一共获得26回忆。

Problem L. 扩散模型

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 256 megabytes

在无监督图像生成隐扩散模型中，一个从标准高斯分布采样得到的噪声隐表示会被逐步去噪，最终得到一个位于变分自编码器隐空间的低维数据流形上的一个隐表示点，并被对应的解码器映射回符合训练数据分布的图片。这类似于由某种形式的微分方程建模的粒子扩散运动。

为了简化问题，我们认为一个采样出的噪声隐表示所有可能的扩散路径形成了一棵以该噪声隐表示为根的扩散树， N 为树上的结点数量。在无监督情况下，每一步去噪都会使隐表示从当前所在的结点随机移动到某个儿子结点上。在经过若干步去噪后，隐表示最终将到达某个叶子结点，从而得到最终的图片。

在这个问题中，我们会标记 K 个不同叶子结点，来表示我们想要隐表示到达的最终结点。然而，扩散模型去噪的随机性往往会导致其无法到达这些结点，因此不一定能生成我们想要的图片。

为了解决这个问题，文生图隐扩散模型在原始隐扩散模型的基础上，引入了自然语言文本语义对去噪过程的控制。相应地，这可以被简化为，有 M 个候选词，我们可以选择是否使用它们。当选用第 i 个候选词时，该候选词会作用于扩散树的结点 u_i 上，使得当隐表示位于 u_i 上时，会被确定性地去噪而移动到 u_i 的儿子结点 v_i 上。而对于未被候选词作用的结点，隐表示在上面时仍然会随机移动到一个儿子结点。

出于效率上的考量，我们希望选用尽可能少的候选词，使得隐表示从根节点出发最终一定能走到一个被标记的叶子结点。

考虑到实际的扩散模型性质，我们保证扩散树上每个叶子结点的深度一致。

特殊的，在选择候选词时，如果有两个候选词作用于同一结点上且均被选择，输入中靠前的候选词会覆盖靠后者的作用。

Input

每个测试点包含多组数据，第一行首先输入数据组数 T 。

对于每组数据：

- 第一行包含三个整数 N, M, K ($2 \leq N \leq 5 \times 10^5, 0 \leq M, K \leq N$)，分别表示去噪树的结点数量、候选词数量和标记叶子结点的数量。
- 接下来是 N 行输入。对于其中第 i ($1 \leq i \leq N$) 行，会先给出一个整数 k_i ，接着是 k_i 个不同的整数 c_{ij} ($i < c_{ij} \leq N$)，表示扩散树结点 i 的所有儿子结点。保证给出的结构是一棵树，且以结点 1 为根。
- 接下来是 M 行输入。其中第 i ($1 \leq i \leq M$) 行包含两个整数 u_i, v_i ，表示第 i 个候选词作用在结点 u_i 上，会确定性地使隐表示移动到结点 v_i ，保证 v_i 是 u_i 的儿子结点。
- 接下来是 1 行输入，包含 K 个互不相同的整数 $t_1, t_2 \cdots t_K$ ，表示被标记的叶子结点。

保证所有数据中 N 的总和不超过 5×10^5 。

Output

输出 T 行。

对于第 i 组数据，输出一行，包含一个整数 $0 \leq L_i \leq M$ ，表示为了满足要求，从 M 个候选词中最少需要选择 L_i 个候选词。特殊的，如果不存在方案使得隐表示一定能到达被标记的叶子结点，则 $L_i = -1$ 。

Example

standard input	standard output
2	2
7 3 2	-1
2 2 3	
2 4 7	
2 5 6	
0	
0	
0	
0	
1 3	
2 4	
3 5	
4 5	
7 2 1	
2 2 3	
2 4 5	
2 6 7	
0	
0	
0	
0	
1 2	
3 6	
4	

Problem M. 不共戴天

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

夜鹭有时候会以青蛙为食。



咪蛤是一只小坏青蛙。小夜鹭小时候被坏咪蛤欺负过，记在了小本本上。现在小夜鹭长大些了，便开始找机会复仇。这天，小夜鹭锁定了咪蛤所栖息的那条小河。水面上依次排布着 n 片荷叶，编号为 $1, 2, \dots, n$ 。

咪蛤有 m 种跳跃策略，第 i 种是从 a_i 号荷叶上跳到 b_i 号荷叶上，其中 $a_i < b_i$, $a_i \neq a_j$ 且 $b_i \neq b_j$ (对 $i \neq j$)。

小夜鹭有 m 种捕食策略，第 i 种是从 x_i 号荷叶旁飞到 y_i 号荷叶旁，其中 $x_i < y_i$, $x_i \neq x_j$ 且 $y_i \neq y_j$ (对 $i \neq j$)。

如果存在两片荷叶 $i \neq j$ ，咪蛤可以从 i 经过一系列跳跃到达 j ，小夜鹭也可以从 i 经过一系列飞跃到达 j ，那么小夜鹭就有机会在某一天逮住咪蛤了。

小夜鹭和咪蛤不共戴天。为了维护世界和平，请你为他们制定各自的策略，使得咪蛤免于一死，且使得策略的数量 m 最大。当然，小夜鹭之后要是发现了，一定会来啄你的脑袋！

Input

n
 $2 \leq n \leq 10^4$

Output

m
 $a_1 \quad b_1$
 $a_2 \quad b_2$
 \vdots
 $a_m \quad b_m$

$x_1 \quad y_1$
 $x_2 \quad y_2$
 \vdots
 $x_m \quad y_m$

Examples

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