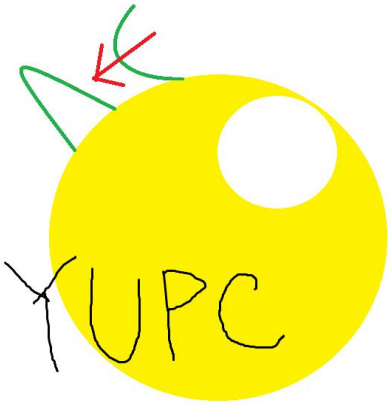
“可爱贤贤杯”第三届柚子程序设计竞赛

半决赛 / Stage 10：广州

第一次竞赛 / Contest 1：荔湾

Based On: Well-Known Problems



Prepared By: HSFZ *NEKO sKORW*

This Page Shoule Be Empty.

不过我啥题都打不好😭

别假了😭

我的 whk 要考不上涟水中专了[流泪]]请使用最新版手机QQ体验新功能

没有 p 啊

豪德

您想要啥服饰可以自己在淘宝上看看，符合 200 以内预算都行

16:18:50

粉兔有啥推荐的吗🥺

没有吧（

依你喜欢

或者依你机房同学喜欢

Problem A. 深搜

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 megabytes

深度优先搜索是一种常见的搜索算法。通过此算法，我们可以从一个无重边、无自环的无向连通图 $G = (V, E)$，和某个出发点 s，得到一棵树 T。

算法的流程描述如下:

1. 将栈 S 设置为空，并令 $T = (V, \emptyset)$，即 T 的边集初始为空。
2. 首先将出发点 s 压入 S 中。
3. 访问栈顶节点 u，并将 u 标记为“已访问的”。
4. 如果存在与 u 相邻且未被访问的节点，则**任意地**从这些节点中挑选一个记为 v。我们将边 (u, v) 加入 T 的边集中，并将 v 压入栈 S 中，然后**回到步骤 3**。若不存在这样的节点，则从栈中弹出节点 u。

可以证明，当图 G 为连通图时，该算法会得到图的某一棵生成树 T。但**算法得到的树 T 可能不是唯一的，它取决于搜索的顺序，也就是算法的第 4 步所选取的顶点**。指定出发点 s 后，如果能够选取一种特定的搜索顺序，使得算法得到的树恰好是 T，则我们称 T 是 G 的一棵 s-dfs 树。

现在给定一棵 n 个顶点的树 T，顶点编号为 $1 \sim n$，并额外给出 m 条边。我们保证这 m 条边两两不同，连接不同的顶点，且与 T 中的 $n-1$ 条树边两两不同。我们称额外给出的 m 条边为**非树边**。在这 n 个顶点中，我们指定了恰好 k 个顶点作为**关键点**。

现在你想知道，有多少种选取这 m 条非树边的方法（可以全部不选），使得：将 T 的边与被选中的非树边构成图 G 之后，存在某个**关键点** s，使得 T 是 G 的一棵 s-dfs 树。

由于答案可能十分巨大，你只需要输出方案数在模 $(10^9 + 7)$ 意义下的值。

Input

输入的第一行包含一个整数 c，表示测试点编号。$c = 0$ 表示该测试点为样例。

输入的第二行包含三个正整数 n, m, k，分别表示顶点个数，非树边的数量，关键点的数量。

接下来 $n-1$ 行，每行包含两个正整数 u, v 表示树 T 的一条边。保证这 $n-1$ 条边构成了一棵树。

接下来 m 行，每行包含两个正整数 a, b 表示一条非树边。保证 (a, b) 不与树上的边重合，且没有重边。

输入的最后一行包含 k 个正整数 $s_1, s_2, ..., s_k$，表示 k 个关键点的编号。保证 $s_1, s_2, ..., s_k$ 互不相同。

Output

输出一行包含一个非负整数，表示方案数在模 $(10^9 + 7)$ 意义下的值。

Example

Standard Input	*Standard Output*
0 4 2 2 1 2 2 3 3 4 1 3 2 4 2 3	3

Note

对于所有测试数据保证：$1 \le k \le n \le 5 \times 10^5$，$1 \le m \le 5 \times 10^5$。

测试点编号	$n \le$	$m \le$	$k \le$	特殊性质
$1 \sim 3$	6	6	n	无
$4 \sim 6$	15	15	6	无
$7 \sim 9$	300	300	6	无
$10 \sim 11$	300	300	n	A
$12 \sim 13$	300	300	n	B
$14 \sim 16$	300	300	n	无
$17 \sim 18$	2×10^5	2×10^5	n	A
$19 \sim 21$	2×10^5	2×10^5	n	B
22	2×10^5	2×10^5	n	无
$23 \sim 25$	5×10^5	5×10^5	n	无

特殊性质 A：保证在 T 中，i 号点与 $i+1$ 号点相连（$1 \le i < n$）。

特殊性质 B：保证若将 T 的边与所有 m 条非树边构成一个图 G，则 T 是 G 的棵 1-dfs 树。

请注意，1 号点不一定是 k 个关键点之一。

Problem B. 能量场

Input file:	*standard input*
Output file:	*standard output*
Time limit:	5 seconds
Memory limit:	2048 megabytes

小 K 有 n 个能量场，第 i 个能量场存储 a_i 点能量。

小 K 在能量场之间建立了 n 条不同的双向能量管道，使得能量场两两连通。

对于一条能量管道，它的能量级为两端能量场能量之和。

小 K 对一组 n 个不同能量管道集合的满意度是所有能量管道能量级的乘积。

现在小 K 想知道，对于所有不同的合法的搭建能量管道的方式，满意度的总和是多少。由于小 K 的满意度是一个 [0,998244353) 之间的整数，所以你只需要输出满意度总和对 998244353 取模后的值即可。

两种搭建管道的方式是不同的当且仅当存在至少一条管道连接能量场 i, j，且恰好在其中一种搭建管道的方式中出现。

Input

第一行一个正整数 n，表示能量场数量。

第二行 n 个非负整数 a_i，表示第 i 个能量场存储的能量。

Output

一行，一个非负整数，表示对于所有不同搭建能量管道的方式，满意度的总和，对 998244353 取模。

Example

Standard Input	*Standard Output*
3 1 2 3	60
4 1 2 3 4	8629
7 1 9 1 9 8 1 0	311816897
16 2 0 0 9 0 2 2 8 2 0 0 9 0 8 1 5	871736512

Note

对于所有数据，保证 $3 \leq n \leq 1000$，$0 \leq a_i < 998244353$。

Subtask	$n \leq$	特殊性质	分数
1	3		1
2	7		4
3	24	✓	5
4	12		10
5	18		10
6	20		5
7	23		5
8	24		30
9	50		15
10	200		5
11	500		5
12	1000		5

特殊性质：保证 $\forall i \in [1, n], a_i = 499122177$。

Problem C. 合并书本

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1.5 seconds
Memory limit:	512 megabytes

小 C 有 n 本书，每本书都有一个重量，他决定把它们合并成一摞。

每一次合并小 C 可以把一摞书放到另一摞书上面，使得它们合并到一摞。如果小 C 把第 i 摞书放到第 j 摞书上面，小 C 需要消耗的体力为**第 i 摞书的重量**加上**两摞书的磨损值之和**。

初始时每本书自成一摞且磨损值均为 0。每当小 C 将两摞书合并后，形成的新的一摞书的磨损值为合并前的两摞书的磨损值的**较大值的两倍再加一**，重量为合并前的两摞书的**重量之和**。

你的任务是设计出合并的次序方案，使小 C 耗费的体力最少，并输出这个最小的体力耗费值。

Input

本题有多组测试数据。

输入的第一行包含一个正整数 t，表示数据组数。

接下来依次输入每组测试数据，对于每组测试数据：

输入的第一行包含一个正整数 n，表示有 n 本书。

输入的第二行包含 n 个正整数，第 i 个数 w_i 表示第 i 本书的重量。

Output

对于每组测试数据输出一行一个整数，表示将 n 本书合并成一摞需要消耗的最少体力。

Example

Standard Input	*Standard Output*
1 4 1 1 1 1	6

Example Explain

如果小 C 将 4 本书两两合并再将得到的两摞合并成一摞，那么前两次需要消耗的体力值各为 1。第三次将一摞重量为 2 的书放到另一摞上面，两摞书磨损值各为 1，需要消耗的体力为 $2+1+1=4$。

因此如果选择这个方案，小 C 耗费的体力只有 $1+1+4=6$。

可以证明，在上述例子中，6 为最小的体力耗费值。

Notes

对于所有测试数据保证：$1 \le t \le 10$，$1 \le n \le 100$，$1 \le w_i \le 10^9$。

测试点编号	$n \le$	是否有特殊性质
$1 \sim 2$	7	否
3	11	否
4	13	否
$5 \sim 6$	22	否
$7 \sim 8$	28	否
$9 \sim 13$	50	否
14	60	否
15	70	否
16	80	否
$17 \sim 18$	100	是
$19 \sim 20$	100	否

特殊性质：保证 $w_i = 1$。

Problem D. 最长待机

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

精灵程序员小 ω 和 小 $\aleph$ 拥有无限的寿命，因此在写代码之余，它们经常玩一些对抗游戏来打发时间。尽管如此，时间还是太多，于是它们发明了一款专用于消磨时间的游戏：最长待机。

为了了解最长待机的规则，首先要了解精灵们使用的编程语言 Sleep++ 的规则：

- 程序由 n 个函数组成，第 $i(1 \le i \le n)$ 个函数具有种类 e_i 和子函数编号序列 $Q_i = (Q_{i,1}, Q_{i,2}, \cdots, Q_{i,l_i})$。$Q_i$ 可以为空，此时 l_i 为 0。
- n 以及所有的 e_i 和 Q_i 可以由程序员任意给出，但它们需要满足以下所有条件：
  - $n \ge 1$；
  - $\forall 1 \le i \le n$，$e_i \in \{0,1\}$；
  - $\forall 1 \le i \le n$，Q_i 中元素两两不同且均为 $[i+1, n]$ 中的整数；
  - $\forall 2 \le j \le n$，**恰好有一个** $Q_i(1 \le i \le n)$ **包含了** j。
- **调用**函数 $i(1 \le i \le n)$ 时，按顺序执行如下操作：
  - 若 $e_i = 0$，令变量 r_i 为 1；否则程序员需要立即为 r_i 输入一个**正整数值**。
  - 若 Q_i 为空，程序等待 r_i 秒；否则重复以下操作 r_i 次：
    - 按顺序**调用**编号为 $Q_{i,1}, Q_{i,2}, \cdots, Q_{i,l_i}$ 的函数。
- 若一个种类为 1 的函数 j 被调用多次，则其每次调用都需要输入 r_j。
- 我们认为，在函数调用中，除了“等待 r 秒”之外的操作不消耗任何时间，即函数调用、运行和输入都在瞬间完成。因此，一个时刻内程序员可能输入多个数。

可以证明，调用任意一个 Sleep++ 程序的任意一个函数，无论如何设定输入，消耗的时间总是有限的。

“最长待机”的游戏规则如下：

- 小 ω 和 小 $\aleph$ 准备好各自的 Sleep++ 程序并选择各自程序中的一个函数。它们互相知晓对方程序的结构以及选择的函数。
- 在时刻 0，小 ω 和 小 $\aleph$ 同时调用自己选择的函数，游戏开始。
- 在时刻 t（$t \ge 0$），双方可以看到对方在时刻 0 至 $(t-1)$ 输入的所有数字，并相应调整自己在时刻 t 输入的数字，但双方无法得知对方在时刻 t 输入的数字。
- 函数调用先结束的一方输掉游戏，另一方胜利。两个调用同时结束算作平局。

小 ω 和 小 $\aleph$ 都是绝顶聪明的，在它们眼中，如果有一方存在必胜策略，那么这局游戏是不公平的。换言之，双方都不存在必胜策略的游戏是公平的。

小 ω 写了一个 n 个函数的 Sleep++ 程序并进行了 m 次操作，操作有以下两种：

- 操作一：给出 k，将 e_k 修改为 $(1 - e_k)$；
- 操作二：给出 k，与小 $\aleph$ 玩一局“最长待机”，开始时小 ω 会调用自己的函数 k。

小 $\aleph$ 信奉极简主义，它希望对于每一局游戏设计出函数个数最少的程序，使得选择其中某个函数能让这局游戏是公平的。你能帮它求出最少所需的函数个数吗？

可以证明，小 ℵ 总是能设计一个程序并选择其中一个函数，使得游戏是公平的。

Input

输入的第一行包含两个正整数 n, m，表示小 ω 的程序中函数的个数以及操作次数。

接下来 n 行，第 i 行若干个整数，描述小 ω 程序中的函数 i：

- 前两个整数 e_i, l_i 表示函数种类和子函数编号序列长度；
- 接下来 l_i 个整数 $Q_{i,1}, Q_{i,2}, \cdots, Q_{i,l_i}$ 描述子函数编号序列。

接下来 m 行，第 j 行两个整数 o_j, k_j 描述一次操作，其中 $o_j = 1$ 表示操作一，$o_j = 2$ 表示操作二。

Output

对于每个操作二输出一行一个整数，表示小 ℵ 的程序中最少所需的函数个数。

Example

Standard Input	*Standard Output*
3 6 0 2 2 3 0 0 0 0 2 1 1 3 2 1 1 3 1 2 2 1	3 3 1

Example Explain

- 对于前两次游戏，小 ℵ 可以给出与小 ω 完全一致的程序并在游戏开始时调用函数 1。可以证明不存在函数个数更少的方案。
- 对于第三次游戏，小 ℵ 可以给出一个仅包含一个种类为 1 的函数的程序，并在游戏开始时调用函数 1。
    - 在时刻 0，小 ω 输入其程序中的 r_2，小 ℵ 输入其程序中的 r_1。
        - 注意：r 变量在小 ω 和小 ℵ 的程序之间是独立的，不会互相影响。
    - 输入完成后，小 ω 的程序在时刻 $(r_2 + 1)$ 结束，小 ℵ 的程序在时刻 r_1 结束。
    - 由于两人在时刻 0 互不知道对方的决策，不能保证 $(r_2 + 1)$ 和 r_1 的大小关系，故双方均不存在必胜策略，这局游戏是公平的。

Notes

对于所有测试数据，

- $1 \le n \le 5 \times 10^5$，$1 \le m \le 2 \times 10^5$；
- $\forall 1 \le i \le n$，$e_i \in \{0,1\}$，$0 \le l_i < n$；
- $\forall 1 \le i \le n, 1 \le j \le l_i$，$i < Q_{i,j} \le n$；
- $\forall 1 \le i \le n, 1 \le p < q \le l_i$，$Q_{i,p} \ne Q_{i,q}$；
- $\forall 2 \le j \le n$，恰好有一个 $Q_i (1 \le i \le n)$ 包含了 j；

- $\forall 1 \le j \le m$，$1 \le o_j \le 2$，$1 \le k_j \le n$。

测试点编号	$n \le$	$m \le$	特殊性质
$1 \sim 2$	3	24	无
3	80	400	AD
4	80	400	BD
$5 \sim 6$	80	400	D
7	3×10^5	10^5	AD
8	3×10^5	10^5	BD
$9 \sim 10$	3×10^5	10^5	D
11	3×10^5	10^5	A
12	3×10^5	10^5	BC
13	3×10^5	10^5	B
$14 \sim 15$	3×10^5	10^5	C
$16 \sim 17$	3×10^5	10^5	无
$18 \sim 19$	5×10^5	2×10^5	A
20	5×10^5	2×10^5	BC
21	5×10^5	2×10^5	B
$22 \sim 23$	5×10^5	2×10^5	C
$24 \sim 25$	5×10^5	2×10^5	无

特殊性质 A：保证

- 任意时刻 e_1 均为 0；
- $\forall 2 \le i \le n$，$l_i \le 1$；
- 操作二的 k 均为 1。

特殊性质 B：保证

- 操作二的 k 满足当时的 e_k 为 1。

特殊性质 C：保证

- $\forall 2 \le i \le n$，$i \in Q_{\lfloor \frac{i}{2} \rfloor}$；
- $\forall 1 \le i \le n$，序列 Q_i 单调递增。

特殊性质 D：保证

- 操作二不超过 10 个；
- 操作二的 k 均为 1。

Problem E. 虫洞

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 megabytes

E 国有 n 个城市，编号为 1 至 n。为了让城市之间的来往更加便利，E 国的交通部想在 n 个城市间建造一些虫洞。每条虫洞是一条**单向**的从某个城市到另一个城市的通道。允许通道的起点和终点是同一个城市，也允许两个城市之间有多个虫洞连接。

为了区分虫洞的建造时间，交通部给每一条虫洞一个正整数的编号。

我们称一种虫洞的建造方案是**好的**，若它满足如下四个条件：

1. 存在一个非负整数 d 使得每个城市恰好是 d 条虫洞的起点，也恰好是 d 条虫洞的终点。
2. 对于每个城市而言，在以它为起点的虫洞的编号中，1 到 d **恰好**各出现一次。
3. 对于每个城市而言，在以它为终点的虫洞的编号中，1 到 d **恰好**各出现一次。
4. 任意选取一个城市 u 和正整数 $1 \le j_1, j_2 \le d$。设从 u 出发，先经过一次编号为 j_1 的虫洞，再经过一次编号为 j_2 的虫洞，到达城市 v_1。设从 u 出发，先经过一次编号为 j_2 的虫洞，再经过一次编号为 j_1 的虫洞，到达城市 v_2。则条件 $v_1 = v_2$ 必定满足。

特别地，不建造任何虫洞的方案也是好的。

现在，建造师已建造了 mn 条虫洞，且给了它们 $1 \sim m$ 的编号，**此时这样的建造方案是好的**。他想要新建造 kn 条虫洞，并给它们 $(m+1) \sim (m+k)$ 的编号。他必须保证这 $(m+k)n$ 条虫洞形成的建造方案仍然是好的。他想知道有多少种新建造 kn 条虫洞的方法，使得这 $(m+k)n$ 条虫洞形成的建造方案是好的。

由于答案很大，你只需要求出方案数除以 998244353 的余数。

Input

输入的第一行四个非负整数 c, n, m, k，其中 c 表示测试点编号。样例中的 c 表示该样例的数据范围与第 c 个测试点的数据范围相同。

接下来 nm 行，每行三个正整数 u, v, w，表示一条编号为 w 的，起点为 u 号城市，终点为 v 号城市的虫洞。

Output

输出一行整数，表示方案数除以 998244353 的余数。

Example

Standard Input	*Standard Output*
1 4 1 1 1 2 1 2 1 1 3 4 1 4 3 1	8

Example Explain

在该组样例中，已经建造的编号为 1 的虫洞为 $1 \to 2$,$2 \to 1$,$3 \to 4$,$4 \to 3$。为了使 8 条虫洞形成的建造方案是好的，新建造的编号为 2 的虫洞可能有 8 种情形：

1. $1 \to 1$,$2 \to 2$,$3 \to 3$,$4 \to 4$
2. $1 \to 1$,$2 \to 2$,$3 \to 4$,$4 \to 3$
3. $1 \to 2$,$2 \to 1$,$3 \to 3$,$4 \to 4$
4. $1 \to 2$,$2 \to 1$,$3 \to 4$,$4 \to 3$
5. $1 \to 3$,$2 \to 4$,$3 \to 1$,$4 \to 2$
6. $1 \to 3$,$2 \to 4$,$3 \to 2$,$4 \to 1$
7. $1 \to 4$,$2 \to 3$,$3 \to 1$,$4 \to 2$
8. $1 \to 4$,$2 \to 3$,$3 \to 2$,$4 \to 1$

Notes

对于所有测试点，

- $1 \le n \le 2 \cdot 10^3$，$0 \le m \le 10^3$，$1 \le k \le 10^{15}$；
- $1 \le u, v \le n$，$1 \le w \le m$；
- 保证初始建造的 mn 条虫洞构成一个号的建造方案。

测试点编号	n	m	k
$1 \sim 4$	≤ 5	≤ 3	≤ 3
$5 \sim 6$	$\le 2 \cdot 10^3$	$= 0$	$= 1$
$7 \sim 8$	$\le 10^2$	$= 1$	$= 1$
$9 \sim 10$	$\le 10^2$	≤ 10	$= 1$
$11 \sim 14$	$\le 10^2$	≤ 10	$\le 10^3$
$15 \sim 16$	$\le 10^2$	$= 0$	$\le 10^{15}$
$17 \sim 19$	$\le 10^2$	≤ 10	$\le 10^{15}$
$20 \sim 21$	$\le 2 \cdot 10^3$	$\le 10^3$	$\le 10^2$
$22 \sim 25$	$\le 2 \cdot 10^3$	$\le 10^3$	$\le 10^{15}$

本题部分测试点输入规模较大，我们推荐你使用较为快速的读入方式。

Problem F. 树的定向

Input file:	*standard input*
Output file:	*standard output*
Time limit:	6 seconds
Memory limit:	2048 megabytes

给定一棵含有 n 个顶点的树，顶点从 1 到 n 编号，树上第 $i(1 \le i \le n-1)$ 条边连接顶点 u_i 和 v_i。

现在，我们想要给树的每条边一个定向。任何一个定向都可以用一个长度为 $n-1$ 的字符串 $S = s_1 s_2 ... s_{n-1}$ 来描述。其中 $s_i = 0$ 代表第 i 条边定向为 $u_i \to v_i$，否则 $s_i = 1$ 代表第 i 条边定向为 $v_i \to u_i$。

给定 m 个顶点对 (a_i, b_i)，其中 $1 \le a_i, b_i \le n$ 且 $a_i \ne b_i$。

一个**完美定向**定义为：在此定向下，对于任意 $1 \le i \le m$，a_i 不能到达 b_i。

试求在所有完美定向中，所对应的字符串字典序最小的定向。**数据保证存在至少一个完美定向**。

定义字符串 $S = s_1 s_2 ... s_{n-1}$ 的字典序小于 $T = t_1 t_2 ... t_{n-1}$ 若存在一个下标 k 使得 $s_1 = t_1, s_2 = t_2, ..., s_{k-1} = t_{k-1}$ 且 $s_k < t_k$。

Input

输入的第一行包含三个非负整数 c, n, m，分别表示测试点编号，树的点数，顶点对的个数。其中 $c = 0$ 表示该测试点为样例。

接下来 $n-1$ 行，每行包含两个正整数 u_i, v_i 表示树的一条边。保证 $1 \le u_i, v_i \le n$ 且这 $n-1$ 条边构成了一棵树。

接下来 m 行，每行包含两个正整数 a_i, b_i。保证 $1 \le a_i, b_i \le n$ 且 $a_i \ne b_i$。

Output

输出一行包含一个字符串 $S = s_1 s_2 ... s_{n-1}$，表示字典序最小的完美定向所对应的 01 字符串。

Example

Standard Input	*Standard Output*
0 4 2 1 2 2 3 3 4 3 2 1 4	001

Example

Standard Input	*Standard Output*
0 6 8 5 1 2 3 1 2 5 6 4 3 4 3 5 1 6 3 5 4 1 4 5 2 3 6 6 2	10101

Example Explain

在样例 1 中，若 $S = 000$，则该定向中 1 能到达 4（存在路径 $1 \to 2 \to 3 \to 4$），因而不是完美定向。若 $S = 001$，则该定向中 3 不能到达 2，1 不能到达 4，因面是完美定向。故答案为 001。

在样例 2 中，一组完美定向必定满足 4 不能到达 3，5 不能到达 1。故 $s_1 = s_5 = 1$。若 $s_2 = s_3 = 0$，则存在路径 $1 \to 2 \to 3 \to 4$，故 1 可到达 4。故其不是完美定向。因此，所有完美定向必定满足 S 的字典序不小于 10101。且容易验证 $S = 10101$ 时，对应的定向是完美定向。

Notes

对于所有测试数据保证 $2 \le n \le 5 \times 10^5$，$1 \le m \le 5 \times 10^5$，$1 \le u_i, v_i \le n$ 且所有的边构成了一棵树，$1 \le a_i, b_i \le n$ 且 $a_i \ne b_i$。

数据保证存在至少一个完美定向。

测试点编号	n	m	特殊性质
$1 \sim 3$	≤ 15	≤ 50	无
$4 \sim 6$	≤ 300	≤ 300	无
7,8	≤ 400	$= (n-1)(n-2)$	A
9,10	$\le 2\,000$	$\le 2\,000$	B
$11 \sim 14$	$\le 2\,000$	$\le 2\,000$	无
15,16	$\le 10^5$	$\le 10^5$	B
17,18	$\le 10^5$	$\le 10^5$	无
$19 \sim 21$	$\le 2 \times 10^5$	$\le 2 \times 10^5$	无
$22 \sim 25$	$\le 5 \times 10^5$	$\le 5 \times 10^5$	无

- 特殊性质 A：保证 (a, b) 出现在 (a_i, b_i) 中当且仅当 $a \ne b$ 且 a, b 在树上不相邻。
- 特殊性质 B：保证树上编号为 1 的顶点与其他每个顶点均相邻。

Problem G. 新本格魔法少女

Input file:	*standard input*
Output file:	*standard output*
Time limit:	6 seconds
Memory limit:	512 megabytes

给定一个长度 n 的整数序列 $a_1, ..., a_n$；

给定一个由 m 次操作构成的操作序列，操作从 1 开始编号，到 m 结束。操作序列中包含修改操作和求和操作，修改操作给定 l, r, v，将 $a_l, a_{l+1}, ..., a_r$ 修改为 v，求和操作给定 l, r，查询 $\sum_{i=l}^{r} a_i$。

共 q 次查询，每次查询给出 L, R，询问将序列 a 初始化为 0 后，依次进行操作序列中的第 $L, L+1, ..., R$ 次操作，每次求和操作的答案之和。

Input

第一行三个整数 n, m, q；

接下来 m 行，每行 $1, l, r, v$ 或 $2, l, r$ 表示一次操作；

接下来 q 行，每行两个整数 L, R 表示一次查询。

Output

共 q 行，每行一个整数，依次表示每次查询的答案。

Example

Standard Input	*Standard Output*
10 5 4 1 9 10 7 1 7 10 9 2 3 10 1 10 10 1 2 5 10 2 5 1 1 3 4 1 3	64 0 0 36

Notes

对所有数据，满足 $1 \le l \le r \le n$，$1 \le v \le n$，$1 \le L \le R \le m$，$1 \le n, m, q \le 5 \times 10^5$。

对 10% 的数据，$n, m, q \le 10^2$。

对另外 20% 的数据，$n, m, q \le 5 \times 10^3$。

对另外 10% 的数据，每次操作都是求和操作。

对另外 20% 的数据，每次查询满足 $L = 1$。

对另外 20% 的数据，$n, m, q \le 2 \times 10^5$。

对于其余数据，无特殊限制。

Problem H. 染色

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1.5 seconds
Memory limit:	512 megabytes

Alice 非常喜欢二进制，她认为事物只有和二进制有关才是美的。

一天，她奇思妙想了一种图案，并打算在长宽都为 2^n 的网格上画出她心中所想的图案。
网格的格子只有黑色和白色两种，一开始都是白色。

现在 Alice 规定一种绘画操作为：选定一个格子，使它自己和相邻上下左右的网格颜色反转，即原本黑色会变成白色，白色会变成黑色。

Alice 还规定网格的第一行和最后一行相邻，第一列和最后一列也相邻。

现在 Alice 希望你给出一个操作方案或告诉无解。如果有多个方案，输出任意一个即可。

Input

第一行一个正整数 n。

接下来一个 $2^n \times 2^n$ 的矩阵，表示 Alice 所想的图案。其中 0 表示白色，1 表示黑色。

Output

第一行一个数 ans 表示操作次数，或输出 -1 表示无解。

接下来 ans 行，每行一个坐标表示操作位置。其中每一维坐标范围均为 $[0, 2^n - 1]$。

Example

Standard Input	*Standard Output*
2 0 0 1 1 1 0 1 0 0 0 0 0 1 1 1 0	7 0 0 1 0 1 3 2 1 3 1 3 2 3 3

Notes

- 对于 20% 的数据，$n = 2$。
- 对于另外 15% 的数据，$n = 4$。
- 对于另外 15% 的数据，$n = 7$。
- 对于 100% 的数据，$n \leq 11$。

Problem I. 线段树

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	512 megabytes

小 Y 最近学会了如何用线段树维护序列，并支持区间求和的操作。

以下给出本题中线段树的定义。该定义可能和你熟知的线段树有区别。

- 线段树是一种有根的二叉树，其每个节点对应了序列上的一个区间 $[l,r)$，其中根节点对应 $[0,n)$。
- 对于每个节点，若其代表的序列区间 $[l,r)$ 满足 $r-l=1$，则其为叶节点；否则存在整数 $m(l<m<r)$，满足其左儿子代表区间 $[l,m)$，右儿子代表区间 $[m,r)$。
- 线段树的形态取决于每个非叶结点的划分点 m 的选择。
- 在区间求和的问题上，对于序列 $a_0,a_1,...,a_{n-1}$，线段树的每个结点 $[l,r)$ 维护了 $(a_l+a_{l+1}+\cdots+a_{r-1})$ 的值。

小 J 有一个长度为 N 的数组 $A_0,A_1,...,A_{N-1}$，他并不知道 A 中的任何一个数，但是他有一棵线段树维护了 A 的区间和。线段树由 $X_1,X_2,...,X_{N-1}$ 给出，其中 X_i 是线段树先序遍历的第 i 个非叶结点的划分点。

例如，如果 $N=5,X=$[2,1,4,3]，则线段树包含的结点的先序遍历为
$[0,5),[0,2),[0,1),[1,2),[2,5),[2,4),[2,3),[3,4),[4,5)$。

小 J 有 M 个区间 $[L_1,R_1),[L_2,R_2),...,[L_M,R_M)$，他想知道，在所有 2^{2N-1} 个线段树结点的子集中，有多少个子集 S 满足以下条件：

- 如果已知 S 中所有结点维护的值，则每个 $[L_i,R_i)$ 区间的和都能被唯一确定。

例如，如果已知 $[0,1),[1,2)$，就能确定 $[0,2)$ 的和；反过来，如果已知 $[0,1),[0,2)$，也能确定 $[1,2)$ 的和。但如果仅已知 $[0,2),[2,4)$ 则不能确定 $[0,3)$ 或 $[1,2)$ 的和。

由于答案很大，你需要输出答案对 998244353 取模后的值。

Input

输入的第一行包含两个整数 N,M，分别表示数组长度和区间个数。

输入的第二行包含 $N-1$ 个整数 $X_1,X_2,...,X_{N-1}$。

接下来 M 行，每行包含两个整数 L_i,R_i，表示一个区间。

Output

输出一行一个整数表示答案对 998244353 取模后的值。

Example

Standard Input	*Standard Output*
2 1 1 0 2	5
2 1 1 1 2	5
5 2 2 1 4 3 1 3 2 5	193
10 10 5 2 1 3 4 7 6 8 9 0 1 0 2 0 3 0 4 0 5 0 6 0 7 0 8 0 9 0 10	70848

Example Explain

只有当直接知道 [0,2) 的总和或同时知道 [0,1) 和 [1,2) 的总和时才能知道 [0,2) 的总和，因此总的方案数为 $2^2 + 1 = 5$。

Notes

对于所有测试数据：

- $2 \leq N \leq 2 \times 10^5$，
- $1 \leq M \leq \min\{\frac{N(N+1)}{2}, 2 \times 10^5\}$，
- $\forall 1 \leq i \leq N-1, 1 \leq X_i \leq N-1$，
- 保证 X_i 正确描述了一棵线段树，
- $\forall 1 \leq i \leq M, 0 \leq L_i < R_i \leq N$，
- $\forall i \neq j, (L_i, R_i) \neq (L_j, R_j)$。

Problem J. 树形图

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	512 megabytes

给定一个 n 个点 m 条边的**简单有向图** G，顶点从 1 到 n 编号。其中简单有向图的定义为**不存在重边与自环**的有向图。

定义顶点 r 是有向图 G 的根当且仅当对于 $1 \leq k \leq n$，顶点 r 到顶点 k 存在恰好一条**有向简单路径**，其中简单路径的定义为**不经过重复点的路径**。

定义每个点的种类如下：

- 若顶点 r 是图 G 的根，则称顶点 r 为图 G 的**一类点**。
- 若顶点 r 不是图 G 的一类点，且存在一种删边的方案，使得图 G 在删去若干条边后得到的图 G' 满足：所有图 G 中的一类点都是 G' 的根，且顶点 r 也是图 G' 的根，则称顶点 r 为图 G 的**二类点**。
- 若顶点 r 不满足上述条件，则称顶点 r 为图 G 的**三类点**。

根据上述定义，图 G 的每个点都恰好属于一类点，二类点，三类点之一。你需要判断点 $1 \sim n$ 分别属于这三个种类中的哪一种。

Input

本题有多组测试数据。

输入的第一行包含一个非负整数 c，表示测试点编号。$c = 0$ 表示该测试点为样例。

输入的第二行包含一个正整数 t，表示测试数据组数。

接下来依次输入每组测试数据，对于每组测试数据：

输入的第一行包含两个正整数 n, m，分别表示有向图的点数和边数。

接下来 m 行，每行包含两个正整数 u, v，表示一条从 u 到 v的有向边。保证 $1 \leq u, v \leq n$，且给定的有向图 G 不存在重边与自环。

Output

对于每组数据，输出一行包含一个长度恰好为 n 的字符串 s 表示每个点的种类。其中 $s_i = 1$ 表示点 i 为**一类点**，$s_i = 2$ 表示点 i 为**二类点**，$s_i = 3$ 表示点 i 为**三类点**。

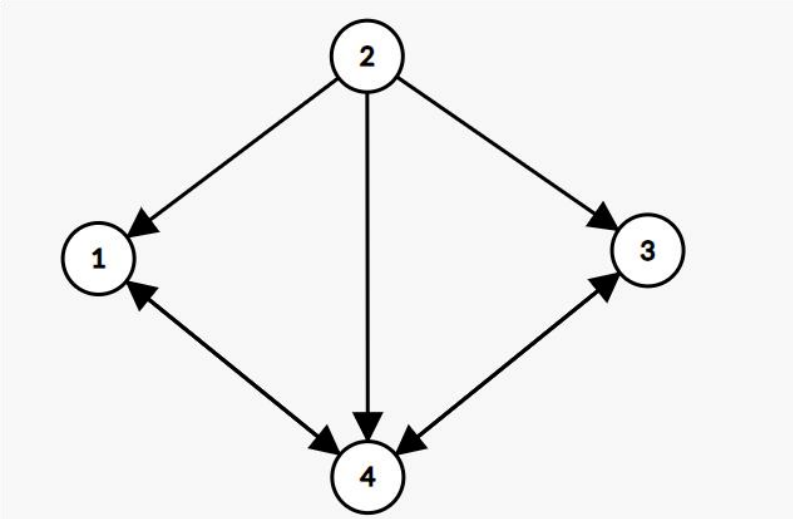
Example

Standard Input	*Standard Output*
0 2 4 7 2 1 4 1 1 4 2 3 3 4 2 4 4 3 4 5 1 2 2 3 2 4 3 1 4 3	3233 2211

Example Explain

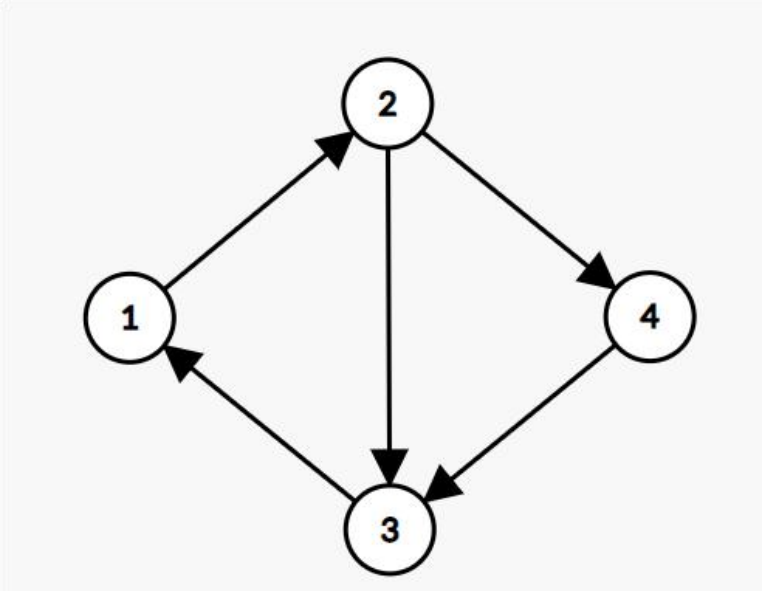
样例 1 共包含两组测试数据。

对于第一组测试数据，输入的图如下：



由于 1,3,4 均不存在到达 2 的路径，因此 1,3,4 均为三类点。由于 2 到 1 的有向简单路径共有三条：$2 \to 1$，$2 \to 4 \to 1$，$2 \to 3 \to 4 \to 1$，因此 2 不是一类点。删去边 $1 \to 4$，$4 \to 1$，$3 \to 4$，$4 \to 3$ 后，2 到 1,3,4 的有向简单路径均唯一，因此 2 是图 G' 的根，即 2 是二类点。

对于第二组测试数据，输入的图如下：



容易发现 3,4 均为一类点，删去边 $2 \to 3$ 后，每个点到其他所有点的有向简单路径均唯一，因此 1,2 均为二类点。

Notes

对于所有测试数据保证：$1 \le t \le 10$，$2 \le n \le 10^5$，$1 \le m \le 2 \times 10^5$，且图 G 不存在重边与自环。

测试点编号	$t \le$	$n \le$	$m \le$	特殊性质
1	3	10	20	无
2	10	10^3	2000	A
3,4	10	10^3	2000	B
5,6	10	10^3	2000	无
7	10	10^5	2×10^5	A
8,9	10	10^5	2×10^5	BC
$10 \sim 13$	10	10^5	2×10^5	B
14,15	10	10^5	2×10^5	C
$16 \sim 20$	10	10^5	2×10^5	无

- 特殊性质 A：保证不存在一类点。
- 特殊性质 B：保证不存在二类点。
- 特殊性质 C：保证编号为 1 的点为图 G 的一类点。

Problem K. 分数

Input file:	*standard input*
Output file:	*standard output*
Time limit:	6 seconds
Memory limit:	512 megabytes

小 Y 和小 C 在玩一个游戏。

定义正分数为分子、分母都为正整数的既约分数。

定义**完美正分数集合** S 为满足以下五条性质的正分数集合：

- $\frac{1}{2} \in S$；
- 对于 $\frac{1}{2} < x < 2$，$x \notin S$；
- 对于所有 $x \in S$，$\frac{1}{x} \in S$；
- 对于所有 $x \in S$，$x + 2 \in S$；
- 对于所有 $x \in S$ 且 $x > 2$，$x - 2 \in S$；

可以证明，上述五条性质确定了唯一的完美正分数集合 S。

所有完美正分数集合 S 中的正分数被称为**完美正分数**。记 $f(i, j)$ 表示 $\frac{i}{j}$ 是否为完美正分数，即 $f(i, j) = 1$ 当且仅当 i 与 j 互素且 $\frac{i}{j} \in S$，否则 $f(i, j) = 0$。

小 C 问小 Y：给定 n, m，求所有分子不超过 n，分母不超过 m 的完美正分数的个数，即求 $\sum_{i=1}^{n} \sum_{j=1}^{m} f(i, j)$。

时光走过，小 C 和小 Y 会再遇见。回首往事，大家都过上了各自想要的生活。

Input

输入的第一行包含两个正整数 n 和 m，分别表示分子和分母的范围。

Output

输出一行包含一个非负整数，表示对应的答案。

Example

Standard Input	*Standard Output*
10 10	16

Example Explain

可以证明，分子分母均不超过 10 的完美正分数共有 16 个，其中小于 1 的 8 个如下：

- $\frac{1}{2}, \frac{1}{4}, \frac{1}{6}, \frac{1}{8}, \frac{1}{10}, \frac{2}{5}, \frac{2}{9}, \frac{4}{9}$。

大于 1 的 8 个完美正分数分别为上述 8 个小于 1 的完美正分数的倒数。

- 可以按照如下方式验证 $\frac{2}{9}$ 是否为完美正分数：因为 $\frac{1}{2} \in S$，$\frac{1}{2}+2=\frac{5}{2} \in S$，$\frac{5}{2}+2=\frac{9}{2} \in S$，$\frac{1}{\frac{9}{2}}=\frac{2}{9} \in S$；
- 可以按照如下方式验证 $\frac{3}{7}$ 是否为完美正分数：假设 $\frac{3}{7}$ 是完美正分数，则 $\frac{1}{\frac{3}{7}}=\frac{7}{3} \in S$，$\frac{7}{3}-2=\frac{1}{3} \in S$，$\frac{1}{\frac{1}{3}}=3 \in S$，$3-2=1 \in S$，与第二条性质矛盾，因此 $\frac{3}{7}$ 不是完美正分数

Notes

对于所有测试数据保证：$2 \leq n, m \leq 3 \times 10^7$。

测试点编号	$n \leq$	$m \leq$
$1 \sim 3$	10^2	10^2
$4 \sim 6$	10^3	10^3
$7 \sim 10$	$8\,000$	$8\,000$
$11 \sim 14$	10^5	10^5
$15 \sim 17$	10^6	10^6
18	8×10^6	8×10^6
19	8×10^6	3×10^7
20	3×10^7	3×10^7

Problem L. Jason 的球

Input file:	*standard input*
Output file:	*standard output*
Time limit:	5 seconds
Memory limit:	512 megabytes

你有 10 个盒子，每个任务会给出一个 n，初始时前 n 个盒子内**可能**有一些球，后 $10-n$ 个盒子为空。

用小写字母表示每种颜色的球，初始时最多只有三种颜色的球，分别用 a,b,c 表示。而在程序中，你可以使用**任何小写字母**表示对应颜色的球。

定义一个盒子**包含**一个字符串，表示对于每种颜色的球，盒子中出现的个数不小于字符串中出现的个数。

从一个盒子中**删除**一个字符串，表示对于字符串中每一个小写字母代表的球，从盒子中取走一个相同颜色的球。删除的前提是需要满足此盒子包含该字符串。

向一个盒子中**放入**一个字符串，表示对于字符串中每一个小写字母代表的每个球，向盒子中放入一个相同颜色的球。

你需要写一个程序完成一些任务，程序包含下面几种语句可供使用：

- change x s y t，其中 x, y 是不超过 10 的非负整数，s, t 是仅由小写字母或单个字符 @ 组成的**非空**字符串（**如果为 @ 则表示将该字符串视作空串**）。如果 x 为 0，则将 k 由 1 遍历到 10，否则 $k = x$。如果盒子 k 中包含字符串 s，从其中删除 s，并在盒子 y 中放入字符串 t，如果 $y = 0$ 则在当前的 k 中（原地）放入，这样就视为成功执行命令。每当成功执行命令后，立刻回到上一个断点，无论 k 是否完全遍历。如果没有成功执行命令，跳转到下一条语句。
- # 表示一个断点。**你必须以一行断点结束整个程序**。认为第 0 行也是一个断点，此断点不计入代价。

语句数可简单地视为程序中 change 和 # 的数量之和，第 0 行的虚拟断点不计入，最后一行的断点计入。

断点数可简单地视为程序中 # 的数量。第 0 行的虚拟断点不计入，最后一行的断点计入。

你不能使用超过 100 条语句，超过 10 的盒子或是非小写字母的球，任意一个盒子中某种颜色的球的个数均不能超过 10^8，程序中的单个字符串长度不能超过 200，你的程序单组数据实际遍历的语句条数不能超过 4×10^5，单组数据实际判断包含的字符集大小之和不能超过 10^7（参考下发的检验器）。

约定 n 表示输入至多使用的盒子数，mx 表示初始时每个盒子中每种颜色球个数的最大值，max 表示初始时所有盒子中球总数的最大值，sum 表示初始时所有盒子中球的总数。**任务中未提及的盒子必须保持原状**，你需要分别完成下面 10 个任务。

1. $n = 10, sum \le 100$，你需要将所有 a 颜色球放入盒子 1，所有 b 颜色球放入盒子 2，所有 c 颜色球放入盒子 3。

2. $n = 10, sum \leq 100$，你需要将所有 a 颜色的球改为 b 颜色的球，将所有 b 颜色的球改为 c 颜色的球，将所有 c 颜色的球改为 a 颜色的球，这三个操作应同时完成。
3. $n = 10, sum \leq 100$，你需要将所有不包含 a 颜色球的盒子清空。
4. $n = 10, sum \leq 100$，你需要给所有非空的，不包含 a 颜色球的盒子放入一个 a 颜色球。
5. $n = 5, sum \leq 5$，你需要将所有球按照颜色从小到大（a < b < c）依次放入盒子 1 到 sum，每个盒子恰好只放一个球，初始的球不保留。
6. $n = 10, sum \leq 100$，对于每个盒子，只保留一个出现次数最多的颜色的球，如果有多种颜色出现次数最多，将其变为空集。
7. $n = 10, sum \leq 100$，对于每个盒子，只保留出现次数最多的颜色的球，如果有多种颜色出现次数最多，保留颜色最小的（a < b < c）。
8. $n = 1, mx \leq 10$，记 a, b, c 颜色的球的个数为 A, B, C，你需要对于满足 $1 \leq x \leq 5$ 的整数 x 使得盒子 x 在最终恰好有 $A + Bx + Cx^2$ 个 a 颜色球，不能有其它颜色球。
9. $n = 5, max \leq 10$，所有 n 个盒子中球总数相等。将每个盒子中的球按从小到大排序，组成一个字符串，保证字符串两两不同。你需要对于每个字符串求出其字典序排名，按字典序从小到大依次将盒子改为 a, b, c, d, e。
10. $n = 5, max \leq 10$，你需要对 n 个盒子求前缀和，即将前面所有盒子的球复制一份放入自己。

注：子任务按某种规则排序，与难度无关。

Format

针对给定的 10 个任务，你需要分别将你的程序命名为 1.out~10.out，并将这 10 个文件直接压缩为 zip 文件提交。

每个文件中需要包含若干行。

第一行一个非负整数 L，代表你使用的语句数。

接下来 L 行，每行一个语句。

你必须以一个断点结尾。

Local Test

第一行，两个整数 T, V，分别表示数据组数与评分参数。

对于接下来每组数据：

第一行，两个整数 n, m，表示需要描述的输入盒子数与输出盒子数。

第二行，n 个字符串，描述输入时前 n 个盒子的状态。

第三行，m 个字符串，描述输出时前 m 个盒子的状态，**你仍然需要保证其它盒子为空**。

同样地，使用 @ 表示空串。

将 checker.cpp 编译后，在命令行执行

checker [in] [out] [ans]

其中 [in] 为测试数据，[out] 为你需要测试的程序，[ans] 输入和 [in] 相同的内容。

例如你需要测试第一个样例，且你的程序名为 1.out，需先将 1.in 复制到当前目录，并执行

checker 1.in 1.out 1.in

Marking Scheme

对于每个测试点，其内部会评测若干组测试数据。

若你的输出出现下列情况，那么该测试点不得分：

- 输出与要求不符。
- 出现无法识别或不合法的语句。
- 某个盒子中某种颜色的球数量超过 10^8。
- 使用超过 100 条语句。
- 程序中单个字符串长度超过 200。
- 使用不在 1 到 10 之间的盒子。
- 使用非小写字母颜色的球。
- 单组数据实际语句遍历次数大于 4×10^5。
- 单组数据实际判断包含的字符集大小之和超过 10^7（参考下发的检验器）。

语句数可简单地视为程序中 change 和 # 的数量之和，第 0 行的虚拟断点不计入，最后一行的断点计入。

断点数可简单地视为程序中 # 的数量。第 0 行的虚拟断点不计入，最后一行的断点计入。

一个程序的代价是你给出的断点数量乘以程序的语句数，记作 val。

否则设对应子任务的评分标准为 V，那么你的得分为：

$$\text{score} = \begin{cases} 11 & V > val \\ \left\lfloor \dfrac{10}{\exp\left(1 - \dfrac{V}{val}\right)} \right\rfloor & \text{otherwise.} \end{cases}$$

下面给出各个任务对应的评分标准 V:

编号	1	2	3	4	5	6	7	8	9	10
V	16	16	16	16	26	8	10	18	20	30

Problem M. Jason 坐电梯

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 seconds
Memory limit:	512 megabytes

一栋 n 层的楼有 m 部电梯，每部电梯有静止与运动两种状态。

初始时，第 i 部电梯静止于第 i 层。给定一个 $1 \sim m$ 的排列 p，你希望最终第 i 部电梯位于 p_i 层。

你可以进行以下两种操作：

- 0：让时间向后运动一个时刻。
- x：其中 x 为不超过 n 的正整数。
    - 执行该操作时，需要满足：x 层不存在静止的电梯；距离 x 层距离最近的 † 静止的电梯存在且唯一。
    - 令 y 为最近的静止的电梯编号，z 为其位置。则电梯 y **立刻**变为运动的电梯，并在 $|x-z|$ 时刻后的**所有操作前**到达楼层 x 并变为静止的电梯。

†：位于 a 层的一部电梯与楼层 x 的距离为 $|a-x|$。

注意：你需要保证，任何时刻不存在两个静止的电梯位于同一楼层。

对于每组数据，有一个评分参数 o，你需要构造出总操作次数不超过 o 的方案才能通过该组数据。

本题使用**自定义校验器**检验你的答案是否正确，因此若有多种满足条件的方案，你只需要输出**任意一种**。

Input

第一行，一个正整数 T，表示数据组数。对于每组数据：

- 第一行，四个正整数 Q, n, m, o，分别表示询问组数、楼层数、电梯数、与评分参数。
- 接下来 Q 行，每行 m 个整数 $p_1, ..., p_m$，表示电梯的目标位置。

Output

对于每组数据：

- 共 $2Q$ 行。对于每个询问，输出两行：
- 第一行，一个非负整数 k，表示你的方案的操作步数；
- 第二行，k 个 $[0, n]$ 中的整数，表示你的具体操作方案。

本题使用**自定义校验器**检验你的答案是否正确，因此若有多种满足条件的方案，你只需要输出**任意一种**。

Example

Standard Input	*Standard Output*
2 2 4 2 12 1 2 2 1 1 10 5 30 5 4 3 2 1	0 9 3 4 0 0 1 0 2 0 0 16 6 6 6 6 6 0 1 0 2 0 3 0 4 0 5 0
1 1 6 5 30 5 4 3 2 1	16 6 6 6 6 6 0 1 0 2 0 3 0 4 0 5 0

Example Explain

对于第一组数据的第一组询问，不需要操作。

对于第一组数据的第二组询问：

操作	时刻	电梯 1 位置	电梯 2 位置
初始状态	0	1	2
3	0	1	运动
4	0	运动	运动
0	1	运动	3
0	2	运动	3
1	2	运动	运动
0	3	4	运动
2	3	运动	运动
0	4	运动	1
0	5	2	1

Notes

对于所有数据，$1 \le T \le 20$，$2 \le m < n \le 5 \times 10^4$，保证 n, m, o 同时满足上述某个子任务的限制，p 为 $1 \sim m$ 的排列，$1 \le Q \le 2 \times 10^6$，$\sum oQ \le 2 \times 10^6$。

子任务	$n \le$	$m =$	$o =$	分值
1	3	2	7	7
2	100	$\lfloor \frac{n}{2} \rfloor$	$2 \times (m+n)$	11
3	40	$n-1$	$3 \times n^3$	17
4	200	$n-1$	$5 \times n^2$	19
5	4000	$n-1$	$50 \times n$	17
6	5×10^4	$n-1$	$6 \times n$	16
7	5×10^4	$n-1$	$5 \times n$	13

Problem N. 鸡

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	512 megabytes

对于一个非负整数序列 a，定义它对应的独立集序列 $f(a)$：

- 假设将 a_i 改为 0，此时选出若干个两两不相邻的数使得它们的和最大，则 $f(a)_i$ 表示和的最大值。

现在给定 n, m，求有多少个长度为 b 的非负整数序列 b 满足以下条件：

- 存在至少一个长度为 n，值域为 $[0, m]$ 的非负整数序列 a 使得 $f(a) = b$。

答案对给定的质数 *MOD* 取模。

Input

共一行，三个数，表示 n, m,*MOD*。

Output

共一行，一个数，表示答案。

Example

Standard Input	*Standard Output*
3 1 1000000007	6
4 2 1000000007	47
20 24 1000000007	901565358
123 234 1000000009	141754844
1234 2345 1004535809	576196526

Notes

对于 100% 的数据，$1 \le n, m \le 3 \times 10^3$，$n \ge 2$，$10^9 < MOD < 1.01 \times 10^9$，*MOD* 为质数。

- Subtask 1（10%）：$n, m \le 5$。
- Subtask 2（15%）：$n \le 300$，$m = 1$。
- Subtask 3（25%）：$n \le 300$，$m \le 5$。
- Subtask 4（20%）：$n, m \le 50$。
- Subtask 5（15%）：$n, m \le 300$。
- Subtask 6（15%）：无特殊限制。

Problem O. 如何正确地排序

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	512 megabytes

有一个 $m \times n$ 的数组 $a_{i,j}$。
定义：

$$f(i,j) = \min_{k=1}^{m}(a_{k,i} + a_{k,j}) + \max_{k=1}^{m}(a_{k,i} + a_{k,j})$$

你需要求出 $\sum_{i=1}^{n}\sum_{j=1}^{n} f(i,j)$。

Input

第一行两个正整数 m, n。

接下来 m 行，每行 n 个正整数表示 $a_{i,j}$。

Output

一行一个正整数，表示答案。

Example

Standard Input	*Standard Output*
3 5 1 7 2 2 7 9 10 4 10 3 7 7 8 10 2	564

Notes

对于所有测试点：$2 \le m \le 4$，$1 \le n \le 2 \times 10^5$，$1 \le a_{i,j} \le 2 \times 10^5$。


おわり
MUSE
Licensed by MUSE