

Problem A. 奇迹

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

Clamee 有一个 $n \times m$ 的矩阵, 他知道一个奇迹值 x 。

在他的矩阵中, 每组 x_1, x_2, y_1, y_2 ($1 \leq x_1 < x_2 \leq n, 1 \leq y_1 < y_2 \leq m$), 如果满足 $a_{x_1, y_1} \oplus a_{x_1, y_2} \oplus a_{x_2, y_1} \oplus a_{x_2, y_2} = x$, 即子矩阵的四角元素的异或值等于 x , 那么就会发生一个奇迹。

Clamee 想让你帮他算算一共有多少次奇迹会发生。

其中 \oplus 表示按位异或运算, 即非负整数按二进制按位进行异或运算, 比如

$$\begin{array}{rcl} 6 & & A6 \\ 1010 & \vdots & 10 \\ 1100 & \vdots & 12 \\ \hline 0110 & \vdots & 6 \end{array}$$

Input

第一行三个数, 依次为 $n, m, (4 \leq n \times m \leq 2 \cdot 10^5), x, (0 \leq x < 2^{20})$,
接下来 n 行, 每行 m 个数, 描述这个矩阵。矩阵元素 $0 \leq a_{i,j} < 2^{20}$

Output

一个非负整数, 表示满足条件的 x_1, x_2, y_1, y_2 的组数。

Example

standard input	standard output
4 3 1 1 4 5 1 4 1 9 1 9 8 1 0	1

Note

样例中, 只有

$$\begin{bmatrix} 9 & 1 \\ 8 & 1 \end{bmatrix}$$

中, $9 \oplus 1 \oplus 8 \oplus 1 = 1$ 。

Problem B. 阶梯计数

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

Imakf 送给了 Clamee 一个 n 行 n 列 ($1 \leq n \leq 10^6$) 的阶梯状棋盘, 为了测试 Clamee 的智商, 他问了 Clamee 一个问题: 在这个棋盘中填 k ($1 \leq k \leq n$) 个相同的棋子, 这 k 个棋子中没有两个棋子在同一行, 也没有两个棋子在同一列的方案数是多少?

智商为 0 的 Clamee 显然回答不上来, 所以 Clamee 想让聪明的你帮他写一个程序算一算方案数对 998244353 取模的结果, 这样他就能在智商测试中及格。

其中 n 行 n 列的阶梯状棋盘是指每行格子左对齐且第 i 行恰有 i 个连续格子的棋盘。

Input

两个数 n, k 。

Output

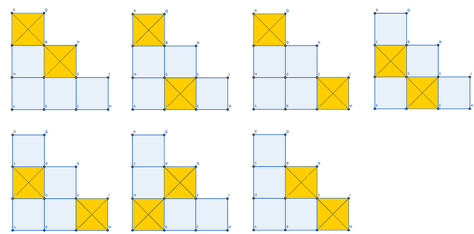
取模后的方案数。

Example

standard input	standard output
3 2	7

Note

以下给出了样例中的 7 种方案:



Problem C. 01感染

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

*Paranoid*总是以为自己写题很厉害, 于是 *ericxie* 给他出了道题。

*Paranoid*现在被给予一个只含0或1的字符串, 除了第一天外, 每天开始时, 与1相邻的0就会变成1。

比如第一天开始时字符串为0010001, 那么第二天开始时字符串就会变为0111011, 第三天开始时就会变为1111111。

现在*Paranoid*要给出 q 次询问的结果, 每次询问都会被给予 d, l, r 三个数字, *Paranoid*要回答出第 d 天时, 01串从第 l 个位置到第 r 个位置范围内, 有多少个字符为1。

注意每天询问是发生在当天的01转化之后的

Input

第一行一个整数 T , 表示数据组数

每组测试数据的第一行是: 第一天的01串 $str(1 \leq \text{len}(str) \leq 200000)$, 查询次数 $q(1 \leq q \leq 200000)$

接下来 q 行, 每行有三个整数 $d(1 \leq d \leq 10^9), l, r(1 \leq l \leq r \leq n)$, 代表题目描述中的意思。

保证 $\sum_{i=1}^T q_i \leq 1000000, \sum_{i=1}^T \text{len}(str)_i \leq 1000000$

Output

每组数据输出 q 行, 代表查询的结果

Example

standard input	standard output
1	1
0010001 3	2
1 1 3	3
2 1 3	
3 1 3	

Problem D. 账号 -1

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

Ysuperman 是一位热爱编程的幼儿园老师, 他为了让他的学生们能够网上学习和练习编程, 开发了一款在线评测系统 (Online Judge)。这个系统可以让小朋友们在网上提交自己写的代码, 然后系统会自动检查代码的正确性和效率, 给出相应的评分和反馈。这样, 小朋友们就可以在玩乐中学习编程, 提高自己的逻辑思维和创造力。

但是这个系统也给 Ysuperman 带来了一些麻烦, 那就是有些无聊的小朋友总是想要找到他的账号, 给他发一些奇怪的私信, 比如问他一些无关紧要的问题, 或者向他吐槽自己的生活。这些私信让 Ysuperman 很烦恼, 因为他觉得这些小朋友应该把时间和精力用在学习编程上, 而不是浪费在这些无聊的事情上。而且, 这些私信也打扰了他的工作和生活, 让他无法专心地开发和维护系统。

为了避免这种情况, Ysuperman 利用自己的特殊权限, 将自己的账号 ID 设置成了 -1 , 这样就不容易被别人发现了。因为一般来说, 账号 ID 都是自然数, 而 -1 这个数字是一个不大于等于 0 的数, 也就是负数。所以, 除非小朋友们知道 Ysuperman 的账号 ID 是 -1 , 否则他们很难找到他的账号。

但是这样做也有一个缺点, 就是有时候小朋友们又需要找到他的账号, 因为 Ysuperman 会经常通过自己的账号发布一些训练题单, 帮助小朋友们提高编程水平。这些题单都是 Ysuperman 精心挑选和设计的, 涵盖了各种编程知识和技巧, 非常适合幼儿园的小朋友们学习。这样一来, 小朋友们就不得不在系统中搜索 -1 这个数字, 才能找到他的账号。这无疑造成了巨大的不便, 让 Ysuperman 很苦恼。

为了解决这个问题, Ysuperman 决定再开发一个新功能, 那就是关注功能。通过这个功能, 小朋友们可以关注自己喜欢的账号, 这样就可以在自己的主页上看到他们发布的题单和动态。Ysuperman 希望通过这个功能, 让小朋友们更方便地找到他的账号, 并且增加系统的互动性和趣味性。

目前系统中有 $n+1$ 个账号, ID 分别为 $-1, 1, 2, \dots, n$ 。其中账号 -1 是 Ysuperman 的账号, 其他的账号 i 是小朋友 i 的账号。最开始只有账号 1 关注了账号 -1 , 且账号 1 只关注了账号 -1 没有关注其他人, 因为小朋友 1 是 Ysuperman 的得意门生。而其它所有的人都没有关注任何人, 因为他们还不知道这个功能的存在。

然后在接下来的 t 天里, 每天都会有一位无聊的小朋友 x_i 闲来无事, 想要看看其他小朋友都关注了谁。于是他就点开了小朋友 y_i 的关注列表, 发现里面有一些他不认识或者不喜欢的人。为了表示自己的不屑或者好奇心, 他就把里面的所有人都关注了一遍。特别地, 如果一个人已经关注了某个人, 再次关注不会有任何效果; 如果一个人试图关注自己, 也会被系统拒绝。

Ysuperman 想知道 t 天之后, 他的账号被哪些小朋友关注了。因为他想了解一下他的系统的使用情况, 以及小朋友们对他的题单和动态的反馈。他希望能够通过关注功能, 和小朋友们建立更好的沟通和联系。他也想知道有多少小朋友是真正喜欢编程的, 而不是只是为了玩。

Input

第一行两个整数, n ($1 \leq n \leq 100$) 和 t ($0 \leq t \leq 1000$)。

接下来 t 行中, 第 i 行有两个整数 x_i 和 y_i ($x_i \neq y_i$ 且 $x_i, y_i \in \{-1, 1, 2, \dots, n\}$)。

Output

第一行一个正整数, s ($1 \leq s \leq n$) 表示 t 天后有几个人关注了 Ysuperman。

第二行 s 个整数, 从小到大输出所有关注了 Ysuperman 的账号的 ID。

Examples

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

Note

第二个样例中:

第一天, 账号 -1 没有被任何人关注, 因为不能关注自己。

第二天, 账号 1 也没有关注任何人, 因为账号 -1 的关注列表是空的。

Problem E. 来自 2020 的鲜花

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

在 NOIP2020 之前, Clamee 给 Awcotn 寄去了一朵鲜花。如今, 宇宙第一快递公司成功地将这朵花送到了 XTU。Clamee 已记不清这朵花长什么样了, 但他还记得这朵花的制作流程, 现在他把流程告诉你, 他想知道他送给 Awcotn 的花有多少种。这朵花可以看成是一个 $n(3 \leq n \leq 10^7)$ 个点的菊花图(星形图), 这 n 个节点从 1 到 n 编号, 它是按以下方法制作的:

菊花图一开始所有节点都未被染色。

Clamee 有 $n - 1$ 种颜料, 他为每个叶子节点指定了一种颜料。

然后他进行了多次(可以是任意次)操作, 每次操作是以下两种之一:

1. 选择一个叶子节点, 将它染成它所对应的颜料的颜色,
2. 选择两个相邻的节点 u, v , 将其中 u 节点染成 v 节点的颜色, 该操作需要保证 v 节点已被染色。

请输出 Clamee 可以生成的合法菊花图总数, 答案对 $10^9 + 7$ 取模。

两张菊花图不同当且仅当两张菊花图存在一个编号相同的点, 满足这个点在两张图中的颜色不同。

一张菊花图合法当且仅当它的所有节点均被染色, 并且可以由以上的两种染色操作染成。

Input

第一行包括一个数 T , 表示数据组数。

接下来 T 行, 每行一个正整数 n , 表示菊花图的点数。

Output

输出包括 $T(1 \leq T \leq 10^5)$ 行, 为对应的合法菊花图总数对 $10^9 + 7$ 取模后的结果。

Example

standard input	standard output
8	4
3	63
4	856
5	13565
6	246036
7	5082595
8	118303984
9	69315780
10	

Problem F. 无产阶级万岁

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

你是无产阶级战士，并且立志要把资本家挂到路灯上。

现在你打算给底层工人发放福利。但是你不希望工人们因为福利不均而打起来。

这个公司也非常神奇，底层工人只有同时在同一个资本家的支配下才能互相联系，了解到对方的情况。

现在每个底层工人都有一个期望福利，你想满足他们所有的愿望，但是如果分配的福利差距太大的话，可能会引起不满，显然如果无法知晓对方情况的话，就不会打起来。

你现在有两种操作避免这种情况的发生。

- 制定非负数 x 的标准
- 选择一个资本家挂在路灯上，这样会有一些底层员工失去联系。

可惜的是，你只有一个路灯了，求一个最小的非负数 x 。

形式化的，给定一棵 n 个节点的无向的树，只有叶子节点有一个点权 a_i ，不允许两个联通的叶子点权之差大于 x ，你现在有一个删去一个节点的操作（即删去这个点和与其相连的边使其变为一个森林），求最小的 x 。

$$3 \leq n \leq 10^5$$

$$\text{叶子数量} \leq 500$$

$$\text{点权大小即 } a_i \leq 10^5$$

特别的，数据保证1不为叶子节点,您可以放心将他当作根节点，根节点不当作叶子节点。

Input

第一行一个数 n 表示有 n 个底层工人和资本家

第二行有 n 个整数，如果是0则是资本家，如果不是0则代表第 i 号底层工人的期望福利

第三行到第 $n+2$ 行，每行有两个整数 u, v 表示 uv 存在领导和被领导的关系。

Output

一个整数 x

Example

standard input	standard output
10 0 0 37 15 0 22 20 0 4 7 1 2 1 8 2 3 2 4 1 5 5 6 5 7 8 9 8 10	18

Problem G. 兄弟数

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

对于某个数正整数 t ，若存在一个正整数 r ，使得 $t = r \times r$ ，则我们称 t 为 **完全平方数**。

对于某个正整数 x ，存在某个正整数 y ，设 $k = x \times y$ ，满足 k 为完全平方数，且 $L \leq k \leq R$ ，则称 y 为 x 的 **兄弟数**。

现在请你解决下面这个问题：

给出 x, L, R ，定义 y 是 x 的兄弟数，如果 y 存在，输出任意一个 y ，如果 y 不存在，则输出 -1 。

Input

第一行输入一个正整数 T ，表示共有 T 组测试样例。

接下来输入 T 行，每行三个正整数 x, L, R 。

$1 \leq T \leq 10^4, 1 \leq x \leq 10^8, 1 \leq L \leq R \leq 10^{16}$

Output

对于每一组测试样例，输出一个正整数，若存在 y 是 x 的兄弟数，则输出 y ，若不存在，则输出 -1 。

Example

standard input	standard output
3	48
27 954 8653	9
49 351 1294	-1
65 754 1533	

Problem H. 蟹黄堡和海霸糊

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

蟹堡王和海之霸的战争不断持续，但是他们也有需要共同解决问题的时候。

蟹黄堡制作了 x 份，海霸糊制作了 y 份，但是很不幸，在计数的时候是把他们放在一起计数的，也就是说，你只知道他们的总和是 n 。

当然，蟹黄堡和海霸糊混在一起不是什么好事，我们定义他们混在一起的不美味程度为 $k = lcm(x, y)$ ，其中 lcm 表示最小公倍数。

任何比奇堡的居民吃了这样混合起来的肮脏食物都能知道这一坨食物的不美味程度是多少。

现在蟹老板和痞老板会告诉你他们制作的产品一共有多少份 n ，居民会告诉你这一坨食物的不美味程度 k 。你现在需要确定 x 和 y 。

Input

第一行一个正整数 T ($1 \leq T \leq 10^5$)，表示样例组数 对于每组样例，每行两个正整数 n ($1 \leq n \leq 10^9$)， k ($1 \leq k \leq 10^{18}$)。

保证输入数据一定有解。

Output

每组样例输出一行两个正整数 x, y , $x \leq y$ ，它们之间用一个空格隔开。

Example

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

Problem I. 三分图

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

题目背景

二分图又称作二部图，是图论中的一种特殊模型。设 $G = (V, E)$ 是一个无向图，如果顶点 V 可分割为两个互不相交的子集 (A, B) ，并且图中的每条边 (i, j) 所关联的两个顶点 i 和 j 分别属于这两个不同的顶点集 A 和 B ，则称图 G 为一个二分图。

我们也可以说，图 G 为二分图，当且仅当顶点 V 可分割为两个互不相交的子集 (A, B) ，并且同集合中任意两点最短路长度不小于 2。

由此定义三分图如下：设 $G = (V, E)$ 是一个无向图，如果顶点 V 可分割为三个互不相交的子集，并且同集合中任意两点最短路长度不小于 3，则称图 G 为由点集 A, B, C 构成的三分图。

题目描述

现在 *Rascalrabbit* 来到了一片群岛，群岛被分成上述的 A, B, C 三部分。

Rascalrabbit 打算在它们之间修建桥梁，使得构成的图是一个三分图（任意两个岛屿是互不相同的）。

Rascalrabbit 三下五除二就求出了修建桥梁的合法方案数，他现在用这个问题来考验聪明的你。

由于结果太大，仅需输出方案数对 998244353 取模的结果。

Input

第一行一个整数 $T(1 \leq T \leq 3000)$ ，表示数据组数

接下来 T 行，每行三个整数 $a, b, c(1 \leq a, b, c \leq 200000)$ ，分别为三个点集的大小

数据保证 $\sum_{i=1}^T (a_i + b_i + c_i) \leq 1.5 \times 10^7$

Output

每组数据输出一行，表示该组数据的答案

Example

standard input	standard output
3	8
1 1 1	63
1 2 2	813023575
6 2 9	

Problem J. Aythsr 的彩票人生

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

Aythsr 是一名外卖小哥，他收到很多的订单需要送。他所在城市的地图可以看成一幅无向连通图，其中有 n 个节点，编号为 $1 \sim n$ ，有 m 条边。每一笔订单由两个参数 (s_i, t_i) 组成，表示他需要从 s_i 点出发，将外卖送到 t_i 点。Aythsr 可以随意规定他的路线，但是为了追求效率，在每一笔订单中，每条边只能经过一遍。

同时 Aythsr 是一个相信幸运的人，城市的某些路上装配了一些“大乐透”彩票机，Aythsr 想知道在他的每笔订单中，是否有机会买彩票。

注意，每笔订单是独立的，每笔订单的过程只计算从 s_i 出发到 t_i 结束，你无须关注从 t_i 到 s_{i+1} 的过程。

Input

第一行输出一个整数 T ，表示有 T 组数据。

每组数据由以下部分组成：

1. 第一行输入三个整数 n, m, q ， n 代表城市节点的数量， m 代表边的数量， q 代表订单的数量。
2. 接下来 m 行描述道路，三个整数 v_i, u_i, f_i ，表示存在一条 (v_i, u_i) 的道路， f_i 为 1 时表示这条路上装配了“大乐透”彩票机， f_i 为 0 时表示这条路上未装配“大乐透”彩票机。
3. 接下来 q 行描述订单，两个整数 (s_i, t_i) ，表示他需要从 s_i 点出发，将外卖送到 t_i 点。

数据规模约定：

$$n, q \leq 10^5, m \leq 2 \times 10^5, \sum n \leq 10^6, \sum m \leq 2 \times 10^6, \sum q \leq 10^6, f_i \in \{0, 1\}$$

$$v_i, u_i, s_i, t_i \in [1, n]$$

图中可以存在重边，自环

Output

对于每个询问，输出一个字符串：

1. 如果可以买彩票，输出：‘wow!golden legendary!’
2. 如果不可以，输出：‘awsl!’

注意输出的大小写！只有小写字母和叹号、空格

Example

standard input	standard output
1	wow!golden legendary!
7 7 4	awsl!
1 2 1	wow!golden legendary!
1 3 0	wow!golden legendary!
3 2 0	
3 7 0	
3 4 0	
4 5 0	
4 6 0	
1 3	
5 6	
4 7	
6 1	

Problem K. 摩天大楼

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

计划建造 n 座的摩天大楼在一条街道上，他们的高度从左往右将分别是 h_1, h_2, \dots, h_n ,

zy 是包工头，在每个施工阶段， zy 都可以选择一个范围 $[l, r]$ ，然后施工队将在此范围内工作，工作结束后，第 l 座到第 r 座摩天大楼的高度都会增加 1。当每座摩天大楼的高度都达到（不能超过）计划高度时，这 n 座摩天大楼的项目就完成了。

令 zy 头疼的是，这个项目的计划总是在变， zy 是个小丑，所以他请你来帮他处理以下两种操作：

1. $[l, r]$ 区间内的计划高度增加 k ,
2. 查询只建设 $[l, r]$ 区间的计划，即 $[1, l-1], [r+1, n]$ 这一段计划高度均为 0 的情况下,最小操作次数.

Input

第一行两个正整数 n 和 m .

第二行 n 个正整数，第 i 个表示 h_i

接下来 m 行，每一行的第一个数 p 表示操作，如果是 $p = 1$ 就是第一种操作，然后紧接三个整数，分别表示 l, r, k ，如果 是 $p = 2$ 就是第二种操作,紧接着两个整数表示 l, r 。

$1 \leq n, m \leq 10^5, 1 \leq h_i \leq 10^6, 1 \leq l \leq r \leq n, p \in \{1, 2\}$

Output

对于每次操作2，每行输出一个整数

Example

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