

第十八届 吉林省
大学生程序设计竞赛

正式赛

2025 年 5 月 24 日



长春理工大学
Changchun University of Science and Technology

试题列表

A	天才琪露诺的天才计算机
B	三角·初华
C	和和积积和积和积
D	互质
E	Eternal Feather
F	Ever Forever
G	石头剪刀布
H	另一个回文问题
I	黑白染色
J	奇偶游戏
K	最大收益
L	好矩阵

本试题册共 12 题，18 页。

如果您的试题册缺少页面，请立即通知志愿者。

Problem A. 天才琪露诺的天才计算机



琪露诺的算术教室开始了哟！

这是一道交互题。

在 9999 年，天才冰精琪露诺制造了支持 1024 位整数运算（`int1024`）的超级计算机 Cirno9！由于琪露诺是天才，这台机器只支持**整数运算**。Cirno9 共有 8 个通用寄存器（变量），名称为 $a, b, c, d, r_0, r_1, r_2, r_3$ 。

作为来自 2025 年的时空穿越者，你获得了临时使用权限。Cirno9 的前四个寄存器 a, b, c, d 分别写入了四个可被 `int1024` 表示的**正整数** a_0, b_0, c_0, d_0 ，其他四个寄存器 $r_0 \dots r_3$ 被写入了 0。你希望使用 Cirno9 比较两个分数 $\frac{a_0}{b_0}$ 和 $\frac{c_0}{d_0}$ 的大小。

你无法得知 a_0, b_0, c_0, d_0 的具体值，但是你可以向 Cirno9 发起**最多** 6666 次操作请求，根据 Cirno9 的反馈结果进行判断。具体可以执行的命令以及格式见下文“交互协议”。

在被琪露诺释放「完美冻结」前，完成计算吧！

补充说明

由于你生活在只有 `int32` 和 `int64`（和 `int128`）的时代，以下是关于 Cirno9 `int1024` 的额外说明：

- `int1024` 的运算能力与 C/C++ 语言当中的 `int` 类似，`int1024` 支持的运算包括加减乘除（`+` `-` `*` `/`）以及大小比较。同样类似于 C/C++ 语言，整数除法的实现是**向零取整**。例如， $100/3 = 33$ ， $(-5)/2 = -2$ ， $(-10)/(-3) = 3$ 。
- `int1024` 能够表示的数据范围是 $[-2^{1023}, 2^{1023})$ 内的整数。运算过程**不能发生溢出**，即运算结果必须能被 `int1024` 表示，否则 Cirno9 会爆炸。特别地，对于**除法运算**，除数不能为 0，否则 Cirno9 会无法停机。

Input

初始时没有输入内容，你需要向交互器发起询问，交互器会返回结果。

Interaction Protocol

你允许使用 8 个 `int1024` 类型的变量，名称分别为 $a, b, c, d, r_0, r_1, r_2, r_3$ 。对应的字符串分别为 `a b c d r0 r1 r2 r3`。

- a, b, c, d 的初始值分别为正整数 a_0, b_0, c_0, d_0 。

- r_0, r_1, r_2, r_3 的初始值均为 0。

你可以进行如下操作:

1. `op x y z`: 计算 $y \text{ op } z$ 存入 x , 其中 $\text{op} \in \{+, -, *, /\}$ 。
 - 如果操作正常进行, 交互器会返回 `ok`, 否则返回 `err`。
2. `? x y`: 比较 x 和 y 的大小关系。
 - 如果操作正常进行, 交互器会返回 `>` 或者 `=` 或者 `<`, 否则返回 `err`。
3. `! rel`: 报告 $\frac{a_0}{b_0}$ 和 $\frac{a_1}{b_1}$ 的大小关系, 其中 $\text{rel} \in \{>, =, <\}$ 。
 - 如果回答结果正确, 交互器会返回 `ok`, 否则返回 `err`。

注意: 对于操作类型 1 和 2, 涉及的变量名可以存在相同。例如 `+ a a b` 或者 `* r0 r0 r0`。

交互器返回 `err` 有以下几种可能:

- 操作不存在;
- 变量不存在;
- 数学运算中发生了溢出或者除零错误;
- 超过了操作次数限制;
- 最终报告的大小关系不正确。

当交互器返回 `err`, 你将得到 `Wrong Answer` 的评测结果。你应该在读取到 `err` 的结果后, 立即终止程序, 否则可能会得到非预期的评测结果。

你可以发起最多 6666 次操作。回答结果不计入操作次数, 其他操作计入 1 次操作次数。

在输出一操作后, 不要忘记换行以及刷新输出缓冲区。否则, 你将会得到 `Time Limit Exceeded` 的评测结果。你可以用如下方式刷新缓冲区:

- 对于 C++, 使用 `fflush(stdout)` 或者 `cout.flush()`;
- 对于 Pascal, 使用 `flush(output)`;
- 对于 Java, 使用 `System.out.flush()`;
- 对于 Python, 使用 `stdout.flush()`。

Example

standard input	standard output
ok	* r0 a d
ok	* r1 b c
ok	- r0 r0 r1
>	? r0 r2
ok	! >

Note

在这个样例中, 交互器内部的初始值为 $a = 99, b = 999, c = 9, d = 99$ 。我们需要比较 $\frac{99}{999}$ 和 $\frac{9}{99}$ 的大小关系。

1. 操作 1: * r0 a d

- 计算 $a \times d = 99 \times 99 = 9801$, 并将结果存入 r_0 。
- 交互器返回 ok, 表示操作成功。

2. 操作 2: * r1 b c

- 计算 $b \times c = 999 \times 9 = 8991$, 并将结果存入 r_1 。
- 交互器返回 ok, 表示操作成功。

3. 操作 3: - r0 r0 r1

- 计算 $r_0 - r_1 = 9801 - 8991 = 810$, 并将结果存入 r_0 。
- 交互器返回 ok, 表示操作成功。

4. 操作 4: ? r0 r2

- 比较 r_0 和 r_2 的大小关系。由于 $r_0 = 810, r_2 = 0$, 因此 $r_0 > r_2$ 。
- 交互器返回 >。

5. 操作 5: ! >

- 由于 $\frac{a_0}{b_0} = \frac{99}{999} \approx 0.0991, \frac{c_0}{d_0} = \frac{9}{99} \approx 0.0909$, 因此程序报告的结果正确。
- 交互器返回 ok, 表示答案正确。

题目附件提供了 Cirno9 的参考实现, 选手可用其测试 int1024。

Problem B. 三角 · 初华

在 Ave Mujica 举办的假面舞会上，三个人偶 Doloris, Oblivionis, Mortis 将沿着由月光点组成的轨迹在舞台上移动。Doloris 宣称，这些轨迹象征着她们被命运编织的丝线。当三个人偶构成的三角形的面积达到最大时，象征着被阴影吞没的月亮将重获生机，而三个人偶会获得暂时的生命。



Ave Mujica

舞台可以看作是一个二维平面。第 i 位人偶的轨迹由 n_i 个月光点组成，依次为 $(x_{i,1}, y_{i,1}), (x_{i,2}, y_{i,2}), \dots, (x_{i,n_i}, y_{i,n_i})$ 。每个轨迹上的人偶将从第一个月光点开始，以单位速度匀速移动到下一个点，到达最后一个点后停下。

请计算，从三个人偶同时出发，到她们均到达各自终点的过程中，三个人偶所构成的三角形的最大面积。

Input

第一行包含三个整数 n_1, n_2, n_3 ($n_i \geq 1, n_1 + n_2 + n_3 \leq 10^5$)，分别表示 Doloris, Oblivionis, Mortis 的轨迹上月光点的数量。

接下来的 n_1 行中，第 i 行包含两个整数 x_i, y_i ($-10^6 \leq x_i \leq 10^6, -10^6 \leq y_i \leq 10^6$)，表示 Doloris 轨迹的第 i 个月光点的坐标。

接下来的 n_2 行中，第 i 行包含两个整数 x_i, y_i ($-10^6 \leq x_i \leq 10^6, -10^6 \leq y_i \leq 10^6$)，表示 Oblivionis 轨迹的第 i 个月光点的坐标。

接下来的 n_3 行中，第 i 行包含两个整数 x_i, y_i ($-10^6 \leq x_i \leq 10^6, -10^6 \leq y_i \leq 10^6$)，表示 Mortis 轨迹的第 i 个月光点的坐标。

保证一条轨迹上相邻的两点的坐标不同。

Output

输出一个实数，表示三个人偶所构成的三角形面积的最大值。

如果你的答案的绝对误差或相对误差不超过 10^{-6} ，则将被视为正确。形式地说，假设你的输出为 a ，标准答案为 b ，当且仅当 $\frac{|a-b|}{\max(1, |b|)} \leq 10^{-6}$ 时，你的输出会被接受。

Examples

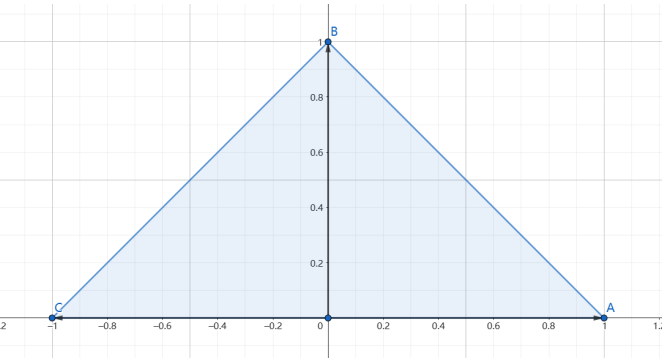
standard input	standard output
2 2 2 0 0 1 0 0 0 0 1 0 0 -1 0	1.0000000000
1 2 2 0 0 0 0 2 2 0 2 2 2	0.3535533906

Note

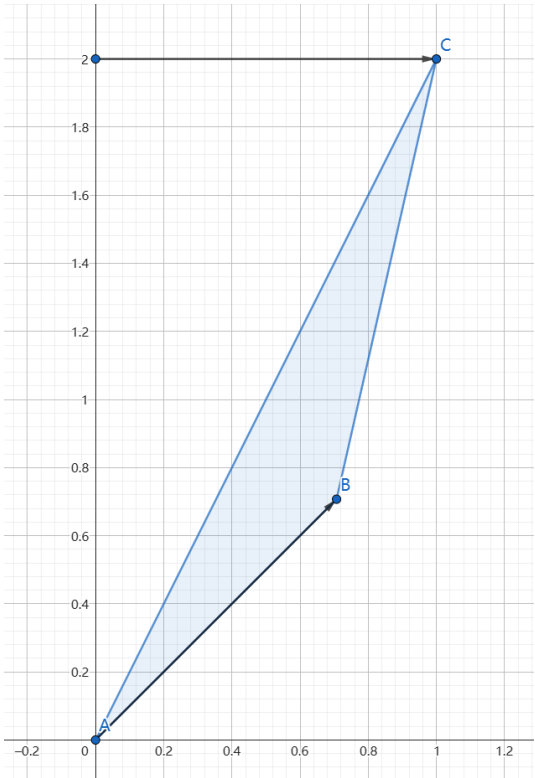
假设三个人偶在 $t = 0$ 时刻同时出发。

对于第一个样例，在 $t = 1$ 时刻三角形面积达到最大。此时 Doloris, Oblivionis, Mortis 的坐标分别为 $(1, 0)$, $(0, 1)$, $(-1, 0)$ ，三角形面积为 1。

对于第二个样例，在 $t = 1$ 时刻三角形面积达到最大。此时 Doloris, Oblivionis, Mortis 的坐标分别为 $(0, 0)$, $(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2})$, $(1, 2)$ ，三角形面积为 $\frac{\sqrt{2}}{4}$ 。

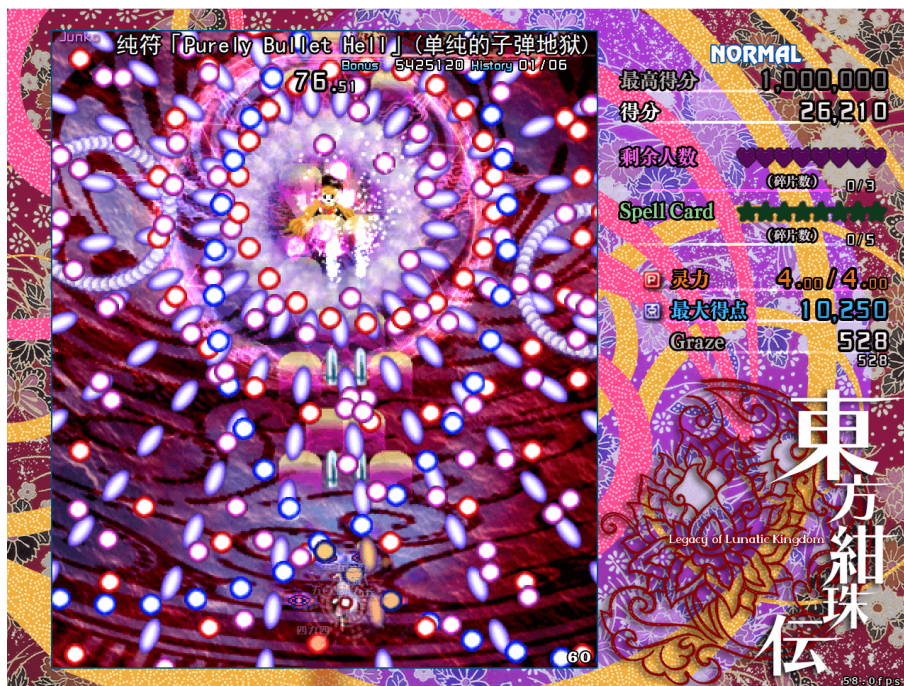


样例一示意图



样例二示意图

Problem C. 和和积积和积和积



小 Wavelet 大战赶猪转

$\Sigma\Pi\Pi\Sigma\Pi\Sigma\Pi\dots$

在博丽神社的后山，灵梦发现了一块刻满奇怪符号的古老石板。即使是熟记幻想乡历史的稗田阿求，面对这些神秘的碑文也一筹莫展。

八云家的式神八云蓝仔细观察后认为，这可能是上古月球文明遗留的碑石，上面记载着某种强大的术式能量公式。

石板上的符号主要由两种运算组成：

- Σ （求和符号）：表示对一系列数值进行相加。例如：

$$\sum_{x=0}^{n-1} f(x) = f(0) + f(1) + \dots + f(n-1)$$

- Π （求积符号）：表示对一系列数值进行相乘。例如：

$$\prod_{x=0}^{n-1} f(x) = f(0) \times f(1) \times \dots \times f(n-1)$$

经过解析，石板上的术式可以表示为：

$$\text{power} = [\text{op}_1]_{x_1=0}^{n-1} [\text{op}_2]_{x_2=0}^{n-1} \dots [\text{op}_k]_{x_k=0}^{n-1} a_{(x_1+x_2+\dots+x_k) \bmod n}$$

其中， $\text{op}_i \in \{\Sigma, \Pi\}$ ，而 a 是一个给定的长度为 n 的能量系数数组。

八云蓝对这个术式的最终能量值感兴趣，但由于计算量过于庞大，即使是擅长计算的她也无法在合理时间内完成。因此她求助于拥有二十一世纪机械式神（计算机）的你。

由于结果可能非常大以至于整个幻想乡都写不下，因此她只需要你给出结果对 998 244 353 取模的值。

Input

第一行包含两个整数 n, k ($1 \leq n, k \leq 10$)。

第二行包含 n 个整数 $a_0, a_1, \dots, a_{n-1}$ ($0 \leq a_i < 998\,244\,353$)。

第三行包含一个长度为 k 的字符串 op , 每个字符为 's' (sum, 代表 Σ) 或 'p' (product, 代表 Π)。

Output

输出一行一个整数, 为计算结果对 $998\,244\,353$ 取模后的值。

Example

standard input	standard output
2 3 1 2 sps	18

Note

对应的数学表达式:

$$\text{power} = \sum_{x_1=0}^1 \prod_{x_2=0}^1 \sum_{x_3=0}^1 a_{(x_1+x_2+x_3) \bmod 2}$$

展开计算过程:

1. 当 $x_1 = 0$ 时,
 - (a) 当 $x_2 = 0$ 时,
 - i. 当 $x_3 = 0$ 时, 结果为 $a_{(0+0+0) \bmod 2} = a_0 = 1$;
 - ii. 当 $x_3 = 1$ 时, 结果为 $a_{(0+0+1) \bmod 2} = a_1 = 2$;
 - iii. 从而内层求和结果为 $1 + 2 = 3$ 。
 - (b) 当 $x_2 = 1$ 时,
 - i. 当 $x_3 = 0$ 时, 结果为 $a_{(0+1+0) \bmod 2} = a_1 = 2$;
 - ii. 当 $x_3 = 1$ 时, 结果为 $a_{(0+1+1) \bmod 2} = a_0 = 1$;
 - iii. 从而内层求和结果为 $1 + 2 = 3$ 。
 - (c) 于是得到中层求积结果为 $3 \times 3 = 9$ 。
2. 当 $x_1 = 1$ 时,
 - (a) 当 $x_2 = 0$ 时,
 - i. 当 $x_3 = 0$ 时, 结果为 $a_{(1+0+0) \bmod 2} = a_1 = 2$;
 - ii. 当 $x_3 = 1$ 时, 结果为 $a_{(1+0+1) \bmod 2} = a_0 = 1$;
 - iii. 从而内层求和结果为 $1 + 2 = 3$ 。
 - (b) 当 $x_2 = 1$ 时,
 - i. 当 $x_3 = 0$ 时, 结果为 $a_{(1+1+0) \bmod 2} = a_0 = 1$;
 - ii. 当 $x_3 = 1$ 时, 结果为 $a_{(1+1+1) \bmod 2} = a_1 = 2$;
 - iii. 从而内层求和结果为 $1 + 2 = 3$ 。
 - (c) 于是得到中层求积结果为 $3 \times 3 = 9$ 。
3. 于是得到外层求和结果为 $9 + 9 = 18$ 。

Problem D. 互质

给定两个正整数 x, y , 保证 $x + 2 \leq y$ 。你需要找到一个正整数 z , 满足:

- $x < z < y$.
- $\gcd(z, y) = 1$ 且 $\gcd(z, x) = 1$.

如果有多个合法的 z , 可以输出任意一个; 如果没有合法的 z , 输出 -1 。

Input

输入的第一行包含一个整数 T ($1 \leq T \leq 10^4$), 表示测试数据的组数。

对于每组测试数据,

仅一行包含两个正整数 x, y ($1 \leq x, y \leq 10^{10}, x + 2 \leq y$)。

Output

对于每组测试数据, 输出一行一个整数, 表示你的答案, 若没有合法答案, 输出 -1 。

Example

standard input	standard output
3	2
1 3	117
115 125	-1
2184 2200	

Problem E. Eternal Feather

Fly high, make it. Get to the new world that I seek. Someday, so I believe. — <euphoric field>

在北半球的一隅，有个地方叫音羽，优子独自栖居于这片土地。她将自己的时光都奉献给了指引那些少女的事业，宛如春风轻拂，温柔地解开他们心中的困惑。

在这漫长的岁月里，她穿梭于无数的问询之中，宛如在时光长河里行舟，直到，她遇见了最后一个问题。就在这关键的时刻，夕充满阴翳的心中突然被一束光照亮，他终于深深领悟到，那个他一直心心念念的少女，始终都在北半球的音羽，带着无尽的温柔与耐心，静静地等待着他。



雨宫优子

给定两个互质参数 p, q , 序列 $f(i)$ 递推式为:

$$\begin{cases} f(0) = 0, \\ f(1) = 1, \\ f(i) = p \cdot f(i-1) + q \cdot f(i-2) \quad (i \geq 2) \end{cases}$$

优子想要计算下列表达式:

$$\sum_{a=1}^n \sum_{b=1}^n \sum_{c=1}^m \gcd(f(c^a + 1), f(c^b + 1))$$

由于答案太大，你只需要输出对 1225 的模数即可。

Input

输入一行，包含四个整数 n, m, p, q ($1 \leq n \leq 10^5, 1 \leq m, p, q \leq 100$)，含义见题目描述。保证 $\gcd(p, q) = 1$ 。

Output

输出一个整数，代表答案。

Examples

standard input	standard output
5 5 3 4	953
3 3 1 1	662
20 10 11 9	333

Note

「Two becomes one, and it through all eternity.」

圣诞快乐，优子。



Ever Forever

Problem F. Ever Forever

Be alive, take it. Promise to find, yes I will. Shining wings filled with wishes. — <euphoric field>



Emotional Flutter

无情的大火吞噬了整个音羽城，在教堂高耸尖塔的阴影下，瑞希的人生轨迹与乐观善良的优子交织在了一起。日子一天天过去，优子身上穷且益坚的意志，和对未来坚定不移的憧憬，为瑞希燃起了久违的温暖。

时光流转，季节更迭，岁月渐渐模糊成遥远的记忆。瑞希发现自己又一次站在了教堂前那象征命运的十字路口。她用微微颤抖的手，温柔地放下了一束鲜花，花瓣仿佛在轻声诉说着早已逝去的往昔。优子曾说过的一句话，在瑞希的脑海中回响，那是时光也无法抹去的旋律。"即便最残酷的命运敲响了门，也要始终带着微笑去面对。"这些话早已成为了瑞希的座右铭，引领着她度过在孤儿院一个个孤寂的夜晚。

瑞希的唇角勾起一抹灿烂的微笑，这笑容仿佛承载着千万个回忆的重量。泪水却在眼眶中打转，随时都可能夺眶而出，无声地证明着，痛苦与思念仍在她心中萦绕不去。那一刻，在过去与现在的交汇中，瑞希仿佛又变回了那个孤苦伶仃的小女孩，在教堂前，哭泣着奔跑着，去拥抱那个支撑自己踟蹰至今的优子姐姐。

瑞希有一个长度为 n 的只包含小写字母的字符串。她想要知道每一对 'e' 与 'f' 的距离之和，其中 'e' 出现在 'f' 之前。形式化的，她想要计算下列和式：

$$\sum_{1 \leq i < j \leq n} [s_i = \text{'e'}] \times [s_j = \text{'f'}] \times (j - i)$$

我们可以回忆到艾弗森括号的计算方法如下：

$$[P] = \begin{cases} 0, & \text{if } P \text{ is false} \\ 1, & \text{otherwise} \end{cases}$$

Input

第一行包含一个整数 n ($1 \leq n \leq 5000$)，表示字符串长度。

第二行包含一个长度为 n ，只由小写字符组成的字符串。

Output

输出一个整数，表示距离之和。

Examples

standard input	standard output
6 efefef	14
25 eternalfeathereverforever	68

Problem G. 石头剪刀布

小 V 和小 Λ 准备玩一个好玩的游戏，石头剪刀布。但传统的石头剪刀布太无聊了，于是他们研发了一款新的游戏，卡牌版石头剪刀布。

具体来说，小 V 和小 Λ 各有 n 张卡片，小 V 有 r_1 张石头， s_1 张剪刀和 p_1 张布，小 Λ 有 r_2 张石头， s_2 张剪刀和 p_2 张布。保证 $n = r_1 + s_1 + p_1 = r_2 + s_2 + p_2$ 。

游戏共持续 n 轮，每轮两位玩家会从自己的手上选择一张卡牌打出，按照石头剪刀布的规则判断胜负，并丢弃掉被打出的两张手牌。如果小 V 获胜累计积分 $+1$ 分，如果小 Λ 获胜累计积分 -1 分，平局则累计积分不变。

现在他们俩想知道，在随机出牌的情况下，累计积分可能的最大值和最小值分别是多少。

Input

输入第一行包含一个整数 T ($1 \leq T \leq 10^4$)，表示测试数据的组数。

对于每组测试数据，

输入一行包含七个整数 $n, r_1, s_1, p_1, r_2, s_2, p_2$ ($0 \leq n \leq 10^9, 0 \leq r_1, s_1, p_1, r_2, s_2, p_2 \leq n, n = r_1 + s_1 + p_1 = r_2 + s_2 + p_2$)，表示游戏的回合数，以及小 V 和小 Λ 的每种卡牌的数量。

Output

对于每组测试数据，输出一行两个整数，分别表示累计积分可能的最大值和最小值。

Example

standard input	standard output
3	3 -3
3 1 1 1 1 1 1	1 1
1 0 1 0 0 0 1	0 0
0 0 0 0 0 0 0	

Problem H. 另一个回文问题

给定一个长为 n 的序列 A , 将 A 序列中的第 i 个元素记为 a_i 。接下来有 q 次操作, 分为修改操作和询问操作:

- 修改操作: 给定一个区间 $[l, r]$ 和一个正整数 v , 将 $a_l, a_{l+1}, \dots, a_r$ 加上 v 。
- 询问操作: 给定一个区间 $[l, r]$, 问序列 $a_l, a_{l+1}, \dots, a_r$ 是否可以通过若干次“交换”形成回文序列, **保证区间长度为偶数**。其中, 对于每次“交换”, 可以选择一个下标 i ($l \leq i \leq r-2$), 交换 a_i 与 a_{i+2} 。

注意, 对于询问操作, 我们只是判断序列能否“交换”形成回文序列, 而不会改变 A 序列。

Input

第一行包含两个整数 n, q ($1 \leq n, q \leq 2 \times 10^5$), 分别表示 A 序列的长度和操作的次数。

第二行包含 n 个整数 $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 10^9$)。

接下来 q 行, 每行表示一个修改或询问操作:

- “0 l r v ” ($1 \leq l \leq r \leq n, 1 \leq v \leq 10^9$), 表示修改操作, 其中 l, r, v 的含义与题目描述相同;
- “1 l r ” ($1 \leq l \leq r \leq n, r-l+1$ 为偶数), 表示查询操作, 其中 l, r 的含义与题目描述相同。

Output

对于每个询问操作, 如果序列可以通过“交换”形成回文, 在一行输出 “YES”, 否则输出 “NO”。你可以以任意形式输出答案 (大写或小写), 比如 “yEs”, “yes”, “Yes” 和 “YES” 都会被认为是肯定的答案。

Examples

standard input	standard output
5 5 2 1 2 1 2 0 2 4 1 1 1 2 1 2 3 1 3 4 1 4 5	YES NO NO YES
9 9 1 2 3 3 2 1 1 2 3 1 1 6 1 4 9 1 2 5 0 1 5 12 1 1 6 1 4 9 0 8 9 12 1 4 9 1 3 8	YES YES YES NO NO YES YES

Problem I. 黑白染色

小 V 和小 A 在玩一个有趣的图论游戏——黑白染色！

游戏在一张 n 个点 m 条边的简单无向图上进行，一开始所有点都没有颜色。游戏开始后，小 V 先进行操作，对于图中的每个点，他可以选择将其染成黑色，染成白色或者不染色。但他的染色方案要满足**不存在一条边，连接的两点都染了色且颜色相同**。随后，小 A 要对图中**所有**还没有颜色的点进行染色，他可以随意将每个点染成黑色或者白色。

最后，两人统计这个图的得分，定义为图中两个端点颜色不同的边的数量。现在小 V 想知道，他有多少种染色方法，使得**无论小 A 如何染色，最后图的得分都不超过 $\frac{m}{2}$** 。由于这个方案可能会很多，所以你只需要输出其对 $10^9 + 7$ 取模的结果。

小 V 的两种染色方法不同，当且仅当存在某点只在其中一种方案中染了色，或者存在某点在两种方案中染的颜色不同。

此外，两个人不喜欢过于稠密的图，因此图中**每个点的度数不超过 3**。

Input

第一行包含一个整数 T ($1 \leq T \leq 10^6$)，表示测试数据的组数。

对于每组测试数据，

第一行包含两个整数 n, m ($1 \leq n, m \leq 10^5$)，表示图的点数和边数。

接下来的 m 行第 i 行包含两个整数 x_i, y_i ($1 \leq x_i, y_i \leq n, x_i \neq y_i$)，表示图中的第 i 条边连接的为点 x_i 和 y_i 。保证图中不存在重边，且**每个点的度数不超过 3**。

保证所有测试中 n 的和以及 m 的和都不会超过 10^6 。

Output

对于每组测试数据，输出一行一个整数，表示小 V 染色的方案数量对 $10^9 + 7$ 取模的结果。

Example

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

Problem J. 奇偶游戏

小 V 和小 A 是数学王国里最要好的朋友。他们经常一起研究各种有趣的数学问题。今天，他们遇到了一个关于奇数和偶数的有趣问题。

小 V 特别喜欢偶数，而小 A 则对奇数情有独钟。他们决定通过一个简单的游戏来决定谁更厉害：每个人选择一个自己喜欢的数字，然后比较它们的大小。

给定两个整数 a 和 b ，保证其中一个数是奇数，另一个数是偶数。你需要根据以下规则输出结果：

- 如果奇数大于偶数，输出 1；
- 如果偶数大于奇数，输出 2。

Input

输入包含两个整数 a 和 b ($1 \leq a, b \leq 10^9$)。

Output

输出一个整数，表示比较结果。

Examples

standard input	standard output
1 2	2
3 2	1

Problem K. 最大收益

小 L 是一家公司的老板，该公司共有 n 个员工，以及 n 个岗位。第 i 个员工的能力值为 a_i ，第 i 个岗位的要求值为 b_i ，第 i 个员工就职第 j 个岗位对公司产生的基础收益为 $a_i + b_j + (a_i \oplus b_j)$ 。其中 $\oplus$ 表示二进制下的异或运算。

同时小 L 发现特定的员工就职特定的岗位会产生额外的效益。小 L 会给出 m 条信息，每条信息形如 (x, y, w) ，即第 x 个员工就职第 y 个岗位会产生 w 的额外收益。

小 L 给出一个参数 K ，他想要知道，对于 $1 \leq k \leq K$ ，若恰好有 k 个员工进行就职，产生的总收益（包括基础收益与额外收益）的最大值为多少？

注意：两个不同的员工不能就职同一个岗位。

Input

第一行包含三个整数 n, m, K ($1 \leq n \leq 10^5, 0 \leq m \leq 5 \times 10^5, 1 \leq K \leq \min(300, n)$)，分别表示员工数/岗位数，信息的条数，所给的参数。

第二行包含 n 个整数，第 i 个整数表示 a_i ($0 \leq a_i < 2^{12}$)。

第三行包含 n 个整数，第 i 个整数表示 b_i ($0 \leq b_i < 2^{12}$)。

接下来 m 行，每行包含 3 个整数 x, y, w ($1 \leq x \leq n, 1 \leq y \leq n, 0 \leq w \leq 10^5$)。

保证不存在两条信息其 x, y 完全相同。

Output

输出一行包含 K 个整数，第 i 个整数表示 $k = i$ 时的答案。

Example

standard input	standard output
5 0 5 1 2 3 4 5 5 4 3 2 1	14 28 42 56 58

Problem L. 好矩阵

对一个大小为 $n \times m$ 的 01 矩阵 A ，如果对于任意 $1 \leq i \leq n, 1 \leq j \leq m$ ，第 i 行和第 j 列的所有元素异或和等于 $A_{i,j}$ （注意 $A_{i,j}$ 会被计算两次），则该矩阵为好矩阵。

形式化地，好矩阵的定义为: $\forall i, j \ (1 \leq i \leq n, 1 \leq j \leq m), \ (\bigoplus_{k=1}^m A_{i,k}) \oplus (\bigoplus_{k=1}^n A_{k,j}) = A_{i,j}$
现给定 n 和 m ，求好矩阵的数量。由于答案很大，你需要输出答案对 998 244 353 取模后的值。

Input

第一行包含一个整数 $T \ (1 \leq T \leq 2 \times 10^5)$ ，表示测试用例的数量。
对于每组测试数据，
输入一行包含两个整数 n 和 $m \ (1 \leq n, m \leq 10^{18})$ ，表示矩阵的行数和列数。

Output

对于每组测试数据，输出一行一个整数，表示答案对 998 244 353 取模后的值。

Example

standard input	standard output
5	16
1 5	2
2 5	64
3 5	16
3 3	400944928
123 456	