

2026 华南师范大学程序设计竞赛 (SCNUCPC 2026)

Magic Team

26 April 2026



Problems

- | | |
|---|---------------------------|
| A | Hello, SCNUCPC! |
| B | 0100101 |
| C | 罪与罚的情人节 |
| D | 情人节摇钱树 |
| E | 学霸题 I |
| F | 学霸题 II |
| G | 关闭核电站 |
| H | 哈集冪乱把路夺 |
| I | 这就是「I」呀 |
| J | 别玩网格图了 |
| K | NTT |
| L | 因为春日将终 |
| M | Until next time, SCNUCPC! |

本试题册共 13 题，24 页。题目顺序与难度无关。
如果您的试题册缺少页面，请立即通知志愿者。

Problem A

Hello, SCNUCPC!

Time limit: 1 second
Memory limit: 256 megabytes



在 8000 年后，全世界的人类与 AI 都在『月读』里打 SCNUCPC。

『月读』中有一条名为『魔法街』的街道。八千代需要布置魔法街，以迎接今年的 SCNUCPC。

魔法街上排列着 n 个摊位，每个摊位的类型可以由一个 $A \sim Z$ 的大写字母表示。

最初街上只有一些摊位布置成 C 型，其他摊位都未布置，标记为 ?。八千代想布置所有未布置的摊位！

摊位的初始布置状态表示为一个长度为 n 的字符串，仅由 C 和 ? 两种字母组成。

用字符串 s 表示布置后的魔法街摊位状态，若 $s_i s_{i+1} \dots s_{i+5} s_{i+6} = \text{SCNUCPC}$ ，则称 i 是一个**赛场**。

请你给出一个布置方案 s ，使得魔法街上恰好有 k 个赛场，或报告无解。

如果存在多种合法方案，输出其中任意一种即可。

Input

每个测试文件由多个测试用例组成。第一行包含一个整数 T ($1 \leq T \leq 10^3$) 表示测试用例数。

对于每个测试用例，第一行包含两个整数 n, k ($0 \leq k < n \leq 2 \times 10^5$)，表示摊位数和赛场数。

随后是一个长度为 n 的字符串，其中仅包含 C 和 ? 两种字符，代表魔法街的初始摊位状态。

数据保证所有测试用例的 n 的总和不超过 2×10^5 。

Output

对于每个测试用例，若无解，输出一行 No；

否则，输出一行 Yes，随后输出一个长度为 n 的字符串，表示布置方案 s 。

Sample

standard input	standard output
2	Yes
7 1	SCNUCPC
????C??	No
16 2	
??C?C??C??C??C?C	

Problem B

0100101

Time limit: 1 second
Memory limit: 256 megabytes

给定一个长度为 n 的 01 串 s 和一个正整数 k 。
你需要构造一个满足以下要求的正整数序列 a :

- 长度为 $n + 1$;
- $1 \leq a_i \leq 100$;
- 对于 1 到 n 之间的任意整数 i , 当且仅当 $s_i = 1$ 时, $\frac{a_i}{a_{i+1}}$ 在 k 进制下数位有限*。

可以证明: 在本题的数据范围约束下, 总有满足以上要求的序列存在。

Input

每个测试文件由多个测试用例组成。第一行包含一个整数 T ($1 \leq T \leq 10^3$) 表示测试用例数。
对于每个测试用例, 第一行包含两个整数 n, k ($1 \leq n \leq 2 \times 10^5, 2 \leq k \leq 10^{18}$)。
第二行包含一个长度为 n 的 01 串 s ($s_i \in \{0, 1\}$)。
数据保证所有测试用例的 n 的总和不超过 2×10^5 。

Output

对于每个测试用例, 输出一行 $n + 1$ 个正整数 a_i ($1 \leq a_i \leq 100$), 表示你构造的序列 a 。

Sample

standard input	standard output
2	2 1 3 6
3 10	1 3 4 5 7 8 9 9
101	
7 2	
0100101	

Note

对于样例的第一个测试用例:

$\frac{a_1}{a_2} = 2, \frac{a_3}{a_4} = \frac{1}{2}$ 均为 10 进制下的整数或有限小数;

$\frac{a_2}{a_3} = \frac{1}{3}$ 为 10 进制下的无限小数。

*例如在 10 进制下, $\frac{1}{8} = 0.125$ 数位有限, 但 $\frac{1}{3} = 0.333\dots$ 数位无限。

Problem C

罪与罚的情人节

Time limit: 1 second
Memory limit: 256 megabytes

因为在情人节偷吃了鼠鼠的奶酪，*Alice* 和 *Bob* 被鼠鼠分别关在两个房间里！

这是一道通信题。

鼠鼠在一个 64×64 的棋盘的每一格上放置了一枚硬币。

每枚硬币要么正面朝上，要么背面朝上。其中有一枚硬币是假的，但它与其他硬币在外观上没有区别。

鼠鼠会先将棋盘拿进 *Alice* 所在的房间，并告诉她哪一枚硬币是假的。

Alice 需要选择棋盘上的一行硬币，将其全部翻面一次，再选择棋盘上的一列硬币，将其全部翻面一次。

随后鼠鼠会将 *Alice* 操作过的棋盘拿进 *Bob* 所在的房间，让 *Bob* 指认假币。

如果 *Bob* 成功指认出假币，两人就能重见天日；否则他们就要遭受惩罚！

现在你需要设计一个程序，在第一次运行时扮演 *Alice*，在第二次运行时扮演 *Bob*，最终指认出假币。

Interaction Protocol

你的程序将会在每次测试中运行两次。

在 *Alice* 阶段，你的程序将扮演 *Alice*；在 *Bob* 阶段，你的程序将扮演 *Bob*。

Alice Round

输入的第一行是单词 *Alice*。

接下来 64 行，每行包含一个长度为 64 的 01 串，表示棋盘上的各枚硬币是否正面朝上。

最后一行包含两个整数 i, j ($1 \leq i, j \leq 64$)，表示棋盘第 i 行第 j 列的硬币是假币。

你需要输出一行两个整数 x, y ($1 \leq x, y \leq 64$)，表示将棋盘第 x 行上的所有硬币翻面一次，再将第 y 列上的所有硬币翻面一次。

如果你的输出不符合格式要求，你将会收到 *Wrong Answer* 的评测结果。

Bob Round

在 *Bob* 阶段，你的程序将被重启。

输入的第一行是单词 *Bob*。

接下来 64 行，每行包含一个长度为 64 的 01 串，表示棋盘上的各枚硬币是否正面朝上。

你需要输出一行两个整数 x, y ($1 \leq x, y \leq 64$)，表示指认棋盘第 x 行第 y 列的硬币为假币。

如果你成功指认出假币，交互器将返回 *Accepted* 的评测结果。

Samples

standard input	standard output
Alice 0101 1010 0101 1010 4 4	2 2
Bob 0001 0001 0001 1110	4 4

Note

受纸张大小限制，样例提供棋盘大小为 4×4 的情形，保证测试数据的棋盘大小均为 64×64 。

在 Alice Round 中，鼠鼠将 4×4 的棋盘递给 Alice，并告诉她第 4 行第 4 列上的硬币是假币。

Alice 选择将第 2 行和第 2 列依次翻面，使得棋盘只有第 4 行和第 4 列上有 3 枚硬币正面朝上。

在 Bob Round 中，鼠鼠将 Alice 进行翻面操作后的棋盘递给 Bob。

Bob 注意到了第 4 行和第 4 列是特别的，指认第 4 行第 4 列上的硬币为假币，通过了鼠鼠的考验！

注意：在实际评测时，棋盘上的硬币状态有 2^{4096} 种可能，上面的规律只是巧合！

Problem D

情人节的摇钱树

Time limit: 1 second
Memory limit: 256 megabytes

在 Alice 和 Bob 逃出生天后，鼠鼠提议：它将再给两人一个谜题。如果解出谜题，他们将获得奖金！有谁会和我过意不去呢？于是两人又被分别关在了两个房间里！

这是一道通信题。

这次 Alice 拿到了一棵节点数为 n 且根节点为 1 的树，一台破旧的电脑，以及一个信号发射器。

Bob 依旧是倒霉的那个，他只拿到了一个信号接收器。

最初电脑屏幕上只有一行 n 个 0，Alice 将利用电脑的键盘对屏幕进行下面的操作：

输入三个数字 l, r, k ，再按一次回车，屏幕的末尾会新增一行 n 个数字，内容复制自上一行，之后新增行的第 l 到 r 个数被修改为 k 。

因为键盘年久失修，所以 Alice 至多只能按 n 次回车键。

此外，Alice 可以利用信号发射器给 Bob 发送一个长度为 n 的排列。

收到来自 Alice 的信号后，Bob 需要利用信号回答 q 次这样的问题：在 Alice 手上的树中，节点 u, v 的 LCA（最近公共祖先）是谁？

对于每次询问，Bob 只需要说出一对数字 i, j ，如果 Alice 的电脑屏幕上第 i 行的第 j 个数恰好是答案，那就算他回答正确。

如果 Bob 能正确回答所有问题，他们二人就能获得奖金！

现在你需要设计一个程序，在第一次运行时扮演 Alice，在第二次运行时扮演 Bob，最终拿到奖金。

Interaction Protocol

你的程序将会在测试中运行两次。

在 Alice 阶段，你的程序将扮演 Alice；在 Bob 阶段，你的程序将扮演 Bob。

Alice Round

输入的第一行是单词 Alice。

第二行包含一个整数 n ($2 \leq n \leq 10^5$)，表示树的节点数。

接下来 $n - 1$ 行，每行包含两个整数 u, v ($1 \leq u, v \leq n$)，表示点 u 和点 v 之间有一条边相连。

你需要先输出一行一个整数 m ($0 \leq m \leq n$)，表示你按回车键的次数。

接下来 m 行，每行输出三个整数 l, r, k ($1 \leq l \leq r \leq n, 1 \leq k \leq n$)，表示输入 l, r, k ，再按一次回车。

最后输出一个长度为 n 的排列* p ($1 \leq p_i \leq n$)。 p 将被用于 Bob 阶段的输入。

如果你的输出不符合格式要求，你将会收到 Wrong Answer 的评测结果。

Bob Round

在 Bob 阶段，你的程序将被重启。

*在本题中指没有元素重复出现的整数序列。

输入的第一行是单词 Bob。

第二行包含两个整数 n, q ($1 \leq q \leq 10^5$), 其中 q 表示询问次数。

第三行是你的程序在 Alice 阶段输出的排列 p 。

接下来会有 q 次询问。每次询问会输入一行两个整数 u, v ($1 \leq u, v \leq n$)。

作为应答, 你需要输出一行两个整数 i, j ($1 \leq i \leq m + 1, 1 \leq j \leq n$), 使得 Alice 的屏幕上第 i 行的第 j 个数恰好是 u, v 在树上的 LCA。

如果你的程序在题目的条件约束内完成 q 次正确应答, 评测机将返回 Accepted 的评测结果。

Samples

standard input	standard output
Alice	1
3	1 3 1
1 2	1 2 3
1 3	
Bob	2 1
3 3	2 2
1 2 3	2 3
1 2	
1 3	
2 3	

Note

在 Alice Round 中, 最初屏幕显示:

0	0	0
---	---	---

随后, Alice 复制上一行, 并将 $[1, 3]$ 内的数字改为 1, 屏幕显示:

0	0	0
1	1	1

最后, Alice 传递了排列 $[1, 2, 3]$ 给 Bob。

在 Bob Round 中, Bob 获得了这个排列。

随后 Bob 回答三个询问, 屏幕对应位置上的数均为 1。

本题附件中提供了一个测试工具以帮助您编写和调试程序。

Problem E

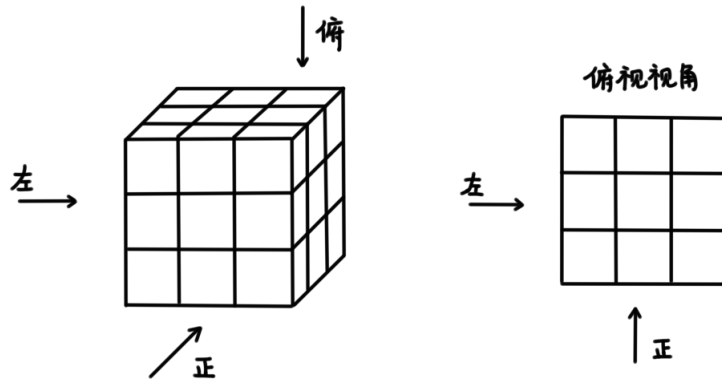
学霸题 I

Time limit: 2 second

Memory limit: 256 megabytes

为了成为学霸，鼠鼠正在学习数正方体。

在一个 $3 \times 3 \times 3$ 的空间里，有一些堆叠起来的边长为 1 的正方体。



鼠鼠通过记录俯视图下每个位置的正方体个数来数这个正方体堆。现在它给你记录的高度矩阵，请你还原出正方体堆的正视图和左视图。

Input

每个测试文件由多个测试用例组成。第一行包含一个整数 T ($1 \leq T \leq 10^5$) 表示测试用例数。

对于每个测试用例，输入一个 3×3 的整数矩阵 $h_{i,j}$ ($0 \leq h_{i,j} \leq 3$)，表示高度矩阵。

Output

对于每个测试用例，输出两个由 * 和 . 组成的 3×3 的字符矩阵，表示正方体堆的正视图和左视图。

其中 * 表示该位置存在正方体，. 表示该位置不存在正方体。

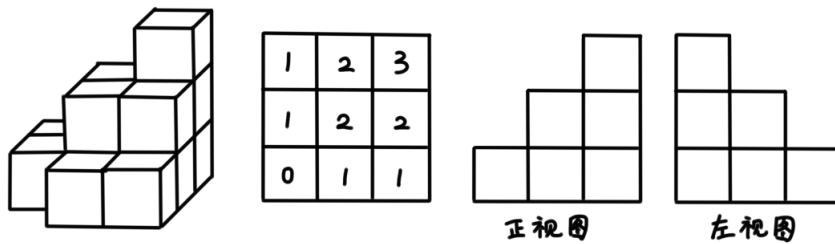
Sample

standard input	standard output
2	..*
1 2 3	.**
1 2 2	***
0 1 1	*..
3 3 3	**.
2 2 2	***
1 1 1	***

	*..
	**.

Note

下图给出了样例一中第一个测试用例对应的正方体堆、高度矩阵、正视图以及左视图。



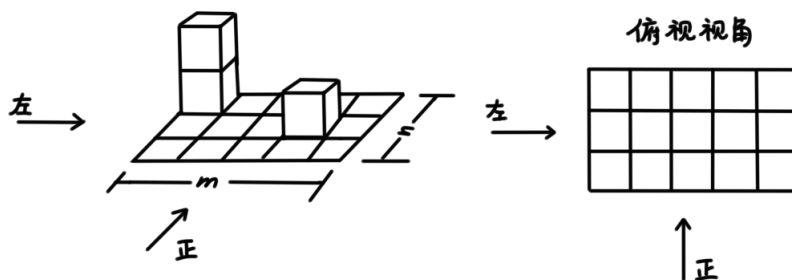
Problem F

学霸题 II

Time limit: 1 second
Memory limit: 256 megabytes

一直在努力成为学霸的鼠鼠会数越来越多的正方体了！它离自己的梦想——考上 *Supreme Cheese National University* 的奶酪统计学专业，越来越近了。

鼠鼠想在地上 $n \times m$ 的区域内堆放一些边长为 1 的正方体。



它设计了一张图纸，上面画着正方体塔的正视图和左视图。为了简便，它只写出该视图下横向每个位置能看到的正方体堆叠高度。

现在你需要告诉鼠鼠是否可能堆出符合图纸的塔。

Input

每个测试文件由多个测试用例组成。第一行包含一个整数 T ($1 \leq T \leq 10^4$) 表示测试用例数。

对于每个测试用例：

第一行包含两个整数 n, m ($1 \leq n, m \leq 2 \times 10^5$)，表示堆叠区域的大小；

第二行包含 n 个整数 a_i ($0 \leq a_i \leq 10^9$)，表示左视图正方体堆叠高度；

第三行包含 m 个整数 b_j ($0 \leq b_j \leq 10^9$)，表示正视图正方体堆叠高度。

保证单个测试文件中，测试用例的 $n \times m$ 之和不超过 2×10^5 。

Output

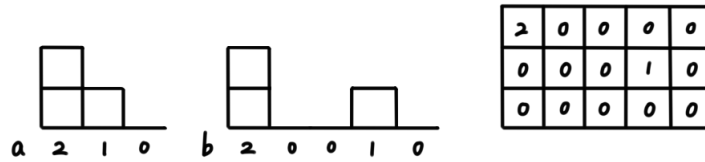
对于每个测试用例，若有解则输出一行 **Yes**；否则输出一行 **No**。

Sample

standard input	standard output
2	Yes
3 5	No
2 1 0	
2 0 0 1 0	
3 5	
0 0 1	
0 1 2 3 0	

Note

对于样例中的第一组数据，对应的正视图、左视图以及一种合法的堆叠方式如下：



对于样例中的第二组数据，可以证明不存在一种合法的堆叠方式。

Problem G

摧毁核电站

Time limit: 2 second
Memory limit: 256 megabytes

准将正在摧毁核电站。

核电站中有 n 个核心，第 i 个核心有能量 a_i 。

核电站中存在 m 条数据线，每条数据线连接两个不同的核心。

为了摧毁核电站，准将需要在核电站内制造足够的干扰。

对于一条连接核心 u, v 的数据线，它会产生数值为 $a_u \text{ XOR } a_v$ 的干扰。

核电站内的干扰是所有数据线产生的干扰之和。

为帮助准将摧毁核电站，金玛丽骇入工厂系统并制作了一个按钮。

每按下一次按钮时，依次执行：

1. 均匀随机生成一个 1 到 n 之间的整数 x ；
2. 均匀随机生成一个 1 到 n 之间的整数 y ；
3. 交换第 x 个核心与第 y 个核心的能量值 a_x, a_y 。

准将想知道，连续按 k 次按钮后，核电站内的干扰期望值是多少。

由于答案可能很大，你只需要输出其对 998244353 取模*后的值。

Input

每个测试文件由多个测试用例组成。第一行包含一个整数 T ($1 \leq T \leq 10^3$) 表示测试用例数。

对于每个测试用例，第一行包含三个正整数 n, m, k ($4 \leq n \leq 2 \times 10^5, 1 \leq m \leq 5 \times 10^5, 0 \leq k \leq 10^{18}$)，分别表示核心个数、数据线条数以及按按钮的次数。

第二行包含 n 个整数 a_i ($0 \leq a_i < 2^{30}$)，表示每个核心的能量。

接下来 m 行，每行包含两个整数 u, v ($1 \leq u, v \leq n, u \neq v$)，表示一条数据线连接核心 u, v 。

数据保证所有测试用例 n 的总和不超过 2×10^5 ， m 的总和不超过 5×10^5 。

Output

对于每个测试用例，输出一行一个整数，表示干扰期望值对 998244353 取模后的值。

*设 $M = 998244353$ ：

- 可以证明，答案能被表示为最简分数 $\frac{p}{q}$ 。
- 存在唯一的整数 r 满足 $0 \leq r < M$ 且 $q \cdot r \equiv p \pmod{M}$ 。
- r 即为你需要输出的数。

Samples

standard input	standard output
1 4 3 1 1 2 4 8 1 2 2 3 3 4	249561110
1 4 6 2 0 2 1 3 4 2 3 4 1 2 2 1 2 4 3 2	62390284

Note

在样例 1 中，按下按钮的两次抽取情况以及对应的核电站干扰值如下表：

	1	2	3	4
1	21	20	18	21
2	20	21	21	27
3	18	21	21	25
4	21	27	25	21

所有情况干扰值总和为 348，干扰值期望为 $\frac{348}{16} = \frac{87}{4} \equiv 249561110 \pmod{998244353}$ 。

Problem H

哈集暴乱把路夺

Time limit: 3 second

Memory limit: 256 megabytes

哈集暴叛乱虽已被平定，但它给魔法王国留下了深远的隐患：王国的大量道路因损毁而难以通行，地方形成割据局势。更严重的是，王国 80% 的孩子都因此染上了 NTT ，这对青少年的危害不可估量！严峻的形势迫使国王考量：叛乱对这个国家的威胁有多大？

魔法王国中有 n 座标号分别为 $1 \sim n$ 的城市与 m 条道路。每条道路连接城市 u, v ，通行难度为 w 。

对于城市 u, v ， $f(u, v)$ 为最小的整数 d ，使得存在若干条通行难度不超过 d 的道路将 u, v 连通。特别地， $f(u, u) = 0$ ；若 u, v 不连通，则 $f(u, v) = +\infty$ 。

对于一组两两连通的城市 S ，如果 $\max_{u, v \in S} f(u, v) < \min_{u \in S, v \notin S} f(u, v)$ ，则称 S 是一个联合。

每个联合 S 具有权重 Γ_S 。最初 $\Gamma_S = \sum_{u \in S} u$ ，即权重为联合内城市标号之和。

任何一个联合都可能宣布独立，但包含同一城市的联合不会同时宣布独立。

若干个联合同时宣布独立称为一个局势，其威胁值为所有宣布独立的联合权重的乘积。

我们认为局势 A, B 不同，当且仅当存在一个联合 S ，在 A, B 恰好一者中宣布独立。

为了更准确地评估威胁，国王会进行 q 次调整。每次调整给定参数 c, k, g 。

对所有包含城市 c 且恰好由 k 座城市组成的联合 $S(c, k)$ ，将 $\Gamma_{S(c, k)}$ 修改为 g 。

在每次调整后，你需要向国王汇报所有可能局势的威胁值之和。

由于答案可能很大，你只需要输出其对 998244353 取模后的值。

Input

每个测试文件由多个测试用例组成。第一行包含一个整数 T ($1 \leq T \leq 10^3$) 表示测试用例数。

对于每个测试用例，第一行包含三个整数 n, m, q ($1 \leq n \leq 10^4, 0 \leq m \leq 2 \times 10^5, 1 \leq q \leq 5 \times 10^5$)，分别表示王国的城市数、道路数以及国王的指令条数。

接下来 m 行，每行包含三个整数 u, v, w ($1 \leq u, v \leq n, 1 \leq w \leq 10^9$)，表示一条道路。

接下来 q 行每行包含三个整数 c, k, g ($1 \leq c, k \leq n, 1 \leq g \leq 10^9$)，表示将 $\Gamma_{S(c, k)}$ 修改为 g 。

数据保证所有测试用例 n 的总和不超过 10^4 ， m 的总和不超过 2×10^5 ， q 的总和不超过 5×10^5 。

Output

对于每个测试用例，在每次调整后输出一行一个整数，代表威胁值之和对 998244353 取模后的结果。

Samples

standard input	standard output
1 3 3 1 1 2 2 1 3 4 2 3 3 1 1 1	41
1 6 5 1 3 2 3 1 4 2 5 3 5 2 1 6 6 4 1 5 2 6	11332

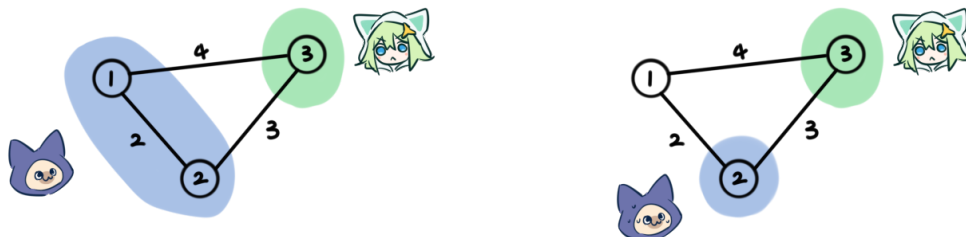
Note

在样例 1 中，联合 [1] 的权重由 1 修改为 1。

所有局势以及对应的威胁值为：

局势	代价	局势	代价
[1]	1	[1 3]	3
[2]	2	[1, 2]	3
[3]	3	[1, 2, 3]	6
[1 2]	2	[1 2 3]	6
[2 3]	6	[1, 2 3]	9

所有局势的威胁值总和为 41。其中两种局势 [1, 2 | 3] 和 [2 | 3] 如下图所示：



Problem I

这就是「I」呀

Time limit: 1 second
Memory limit: 256 megabytes



人生中大家都愚蠢得可爱

在奋力挣扎后迎来雨过天晴没错

这是人类历史上前所未有的光辉

这就是「I」呀——

enana 的一幅画作可以用一个大小为 $n \times m$ 的黑白方格图表示。

她称画作中的一个子矩形为一个「I」，当且仅当其满足如下约束：

- 列数为奇数；
- 行数和列数均不小于 3；
- 中间一列全黑；
- 第一行与最后一行全黑。

未受到约束的方格可以是任意颜色。

现在她需要你帮忙计算出画作中「I」的个数。

Input

每个测试文件由多个测试用例组成。第一行包含一个整数 T ($1 \leq T \leq 10^3$) 表示测试用例数。

对于每个测试用例，第一行包含两个整数 n, m ($3 \leq n, m \leq 2 \times 10^5$)，分别表示画作的行数和列数。

随后是一个大小为 $n \times m$ 的字符矩阵，其由 $*$ 和 $.$ 组成，分别表示画作的黑白格。

数据保证所有测试用例的 $n \times m$ 的总和不超过 2×10^5 。

Output

对于每个测试用例，输出一行一个整数，即画作中「I」的个数。

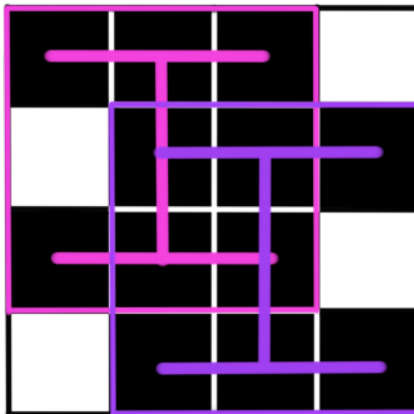
Sample

standard input	standard output
2	2
4 4	4
***.	
.***	
***.	
.***	
3 5	

.***.	

Note

在样例 1 中，一共有 2 个「I」：



Problem J

别玩网格图了

Time limit: 1 second
Memory limit: 256 megabytes

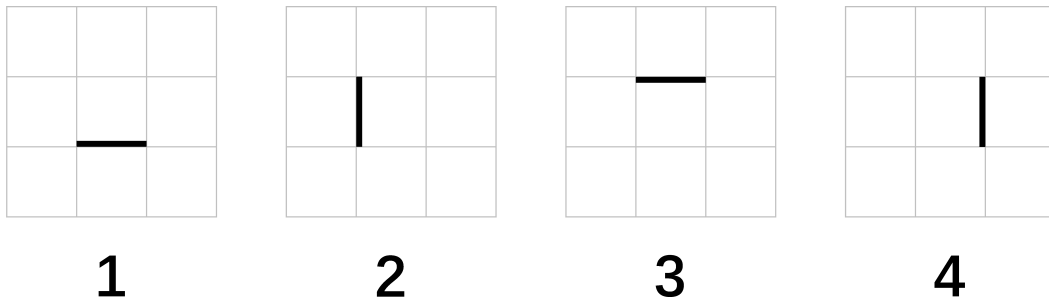
你完成了上次的谜题，又被传送到另一个世界。这次这里没有香草，只有在 *cos* 香草的 *pwp*，坐在空白世界里唯一的桌子上。

你是谁，你怎么穿着品...咳咳，香子兰的衣服？问题还在心里没有说出口，*pwp* 便回答，"*pwp* 就是 *pwp* 喵。奇怪的是，这里明明是君的梦境，怎喵会梦到 *pwp* 喵？"*pwp* 说着，晃着双腿。梦吗？那..."是哦。不过别想着在梦里干什喵怪事喵，心声都露出来了哦。"*pwp* 说着，按了几下桌上的铃。唔...

"更优先的不是应该从梦里出去喵？"*pwp* 指了指地面。又是网格图？"既然从上一层梦境下来了，总应该能破解这个谜题叭喵。"*pwp* 说完，尾巴把桌上一张 +4 牌扫下，落在某个格子边缘，落在的那条边忽地变得更醒目更宽大了。

人类，上面的话都不重要喵——*pwp* 参上 ~

给定 $n \times m$ 大小的网格图，在每个网格放置一个带有一条边的方形：



求用 1, 2, 3, 4 标识对应四种方向的方形，在这个网格图上能围出的最大封闭空间面积。

Input

每个测试文件由多个测试用例组成。第一行包含一个整数 T ($1 \leq T \leq 1500$) 表示测试用例数。

对于每个测试用例，输入一行两个整数 n, m ($1 \leq n, m \leq 10^6, 1 \leq n \times m \leq 10^6$)，表示网格图大小。

数据保证所有测试用例的 $n \times m$ 的总和不超过 10^6 。

Output

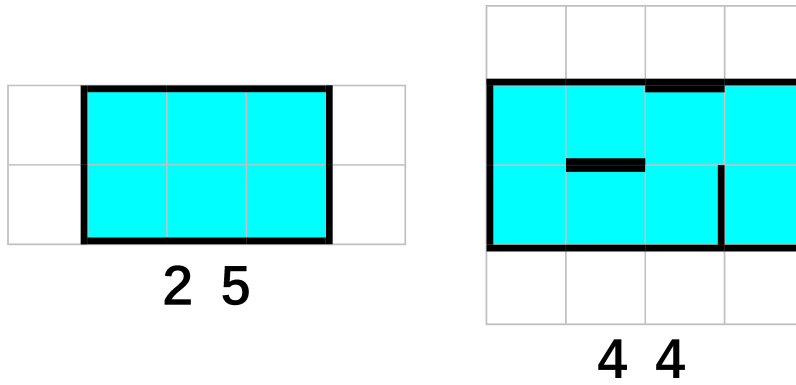
对于每个测试用例，第一行输出一个整数，表示能围出的最大封闭空间的面积；若无法形成面积大于 0 的封闭空间，则输出 0。

Sample

standard input	standard output
2	6
2 5	8
4 4	

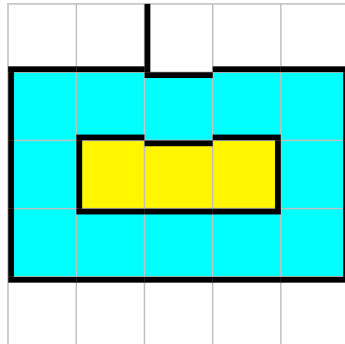
Note

下图为样例解释。



容易证明没有严格更优的答案。

以下部分为消歧义说明，对于如下图的构造方案



同时存在两个封闭空间，面积分别为 12（浅蓝部分）和 3（黄色部分）。

Problem K

NTT

Time limit: 3 second

Memory limit: 256 megabytes



给定一个 n 次多项式函数 $F(x) = f_0 + f_1x + f_2x^2 + \dots + f_nx^n$ ，以及一个素数 p 。

随后有 q 次询问，每次给定一个整数 x_i ，你需要求出 $F(x_i) \bmod p^*$ 的值。

Input

每个测试文件由多个测试用例组成。第一行包含一个整数 T ($1 \leq T \leq 100$) 表示测试用例数。

对于每个测试用例，第一行包含三个整数 n, p, q ($1 \leq n \leq 10^6, 2 \leq p < 5 \times 10^3, 1 \leq q \leq 5 \times 10^3$)，分别表示多项式 $F(x)$ 的次数、给定的素数以及询问的次数。

第二行包含 $n + 1$ 个整数，表示 $F(x)$ 从低次到高次的系数 f_i ($0 \leq f_i \leq 10^6$)。

接下来 q 行，每行包含一个整数 x_i ($|x_i| \leq 10^6$)，你需要求出 $F(x_i) \bmod p$ 的值。

数据保证所有测试用例 n 的总和各不超过 10^6 ， q 的总和不超过 5×10^3 。

Output

对每次询问，输出一行一个整数，即 $F(x_i) \bmod p$ 的值。

*可以证明，存在唯一的整数 r 满足 $0 \leq r < p$ 且 $F(x_i) \equiv r \pmod{p}$ ， r 即为你需要输出的数。

Sample

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

Note

对于样例中第一个测试用例的第二个询问：

$$F(x) = 9x + 6x^3$$

$$F(-7) = 9 \times (-7) + 6 \times (-7)^3 = -2121 \equiv 4 \pmod{5}$$

Problem L

因为春日将终

Time limit: 3 second
Memory limit: 256 megabytes



钱包日渐消瘦，祥子意识到——在 *crychic* 的那些日子已经一去不复返了。抬起用 96 块钱买来的迈巴赫，她再次奔走在泥泞的小路之间...

祥子所在城市的道路结构可以看作一棵包含 n 个节点，以 1 号节点为根的树。

最初每个节点 u 各有工资倍率 a_u 。受战乱影响，少部分（不超过 100 个）节点客流量极不稳定，被称为特殊节点。

接下来 q 天中，特殊节点在第 i 天的工资倍率将统一调整为 p_i 。

记祥子出车时经过节点集合为 T ，则当日工资为 $\prod_{u \in T} a_u$ 。为保证服务质量，祥子在出车时有以下原则：

- 从 1 号节点开始出车。
- 只会沿着树上的边向相邻的节点移动，但可以重复经过同一节点。
- 若出车时经过节点 u ，则本次出车要么不经过 u 的任何儿子节点，要么经过 u 的所有儿子节点。

两个出车方案不同，当且仅当二者经过的节点集合不同。

祥子需要你在第 i 天计算出当天所有合法出车方案的工资之和。

由于答案可能很大，你只需要输出其对 $10^9 + 7$ 取模后的值。

Input

每个测试文件由多个测试用例组成。第一行包含一个整数 T ($1 \leq T \leq 10^3$) 表示测试用例数。

对于每个测试用例，第一行输入两个整数 n, q ($1 \leq n, q \leq 2 \times 10^5$)，分别表示节点数和天数。

第二行包含 n 个整数 a_i ($0 \leq a_i \leq 10^6$)，表示每个节点的初始工资倍率。其中初始倍率为 0 的节点为特殊节点，每个测试用例的特殊点数不超过 100。

接下来 $n - 1$ 行每行包含两个整数 u_i, v_i ($1 \leq u_i, v_i \leq n$)，表示一条边将节点 u_i 与 v_i 连接。

接下来 q 行每行包含一个整数 p_i ($1 \leq p_i \leq 10^9$)，具体含义见题面描述。

数据保证所有测试用例 n, q 的总和各不超过 2×10^5 。

Output

对于每个测试用例，输出 q 行，每行包含一个整数，即当天所有合法出车方案的工资之和对 $10^9 + 7$ 取模后的结果。

Sample

standard input	standard output
1	44
4 3	30
2 0 2 2	58
2 3	
3 4	
2 1	
3	
2	
4	

Note

在样例 1 中，道路结构图、特殊节点、每天的工资倍率情况如下：

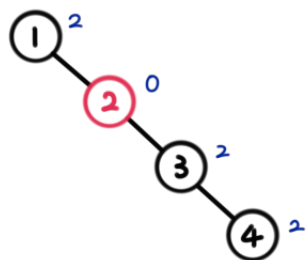


图1 初始

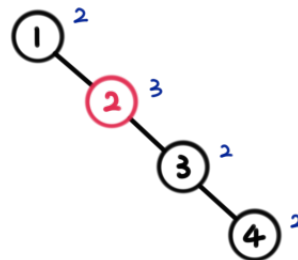


图2 第1天

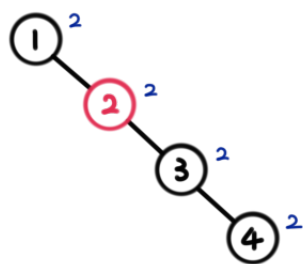


图3 第2天

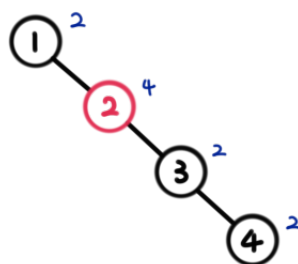


图4 第3天

在第一天中，合法的出车方案以及对应工资：

出车方案	工资
[1]	2
[1, 2]	$2 \times 3 = 6$
[1, 2, 3]	$2 \times 3 \times 2 = 12$
[1, 2, 3, 4]	$2 \times 3 \times 2 \times 2 = 24$

所有方案的工资之和为 $2 + 6 + 12 + 24 = 44$ 。

Problem M

Until next time, SCNUCPC!

Time limit: 1 second
Memory limit: 256 megabytes

在 SCNUCPC 结束后，八千代要将魔法街布置回日常的样子。

魔法街上排列着 n 个摊位，每个摊位只有两种状态：已经布置成 C 型；或尚未布置，标记为 ?。

摊位的初始布置状态表示为一个长度为 n 的字符串，仅由 C 和 ? 两种字母组成。保证第一个摊位和最后一个摊位一定是 C 型摊位。

八千代需要将部分 ? 布置成 C，其余位置仍保持为 ?。

记街上 C 型摊位的位置依次为 p_1, p_2, \dots, p_k ，对任意 $i > 1$ ， p_i 处的 C 型摊位会产生 $p_i \cdot (p_i - p_{i-1})$ 的开摊代价。

请你求出魔法街上恰好有 k 个 C 型摊位时，总开摊代价的最小值。

Input

每个测试文件由多个测试用例组成。第一行包含一个整数 T ($1 \leq T \leq 10^3$) 表示测试用例数。

对于每个测试用例，第一行包含两个整数 n, k ($2 \leq k \leq n \leq 2 \times 10^5$)，表示总摊位数和 C 型摊位数。

随后是一个长度为 n 的字符串，其中仅包含 C 和 ? 两种字符，代表魔法街的初始摊位状态。保证 C 的个数不超过 k 。

数据保证所有测试用例的 n 的总和不超过 2×10^5 。

Output

对于每个测试用例，输出一行一个整数，即总开摊代价的最小值。

Sample

standard input	standard output
2	20
5 2 C????C	31
7 4 CC????C	

Note

在样例的第一个测试用例中，C????C 是唯一方案，开摊代价为 $5 \times (5 - 1) = 20$ 。

在样例的第二个测试用例中，方案 CC?C??C 可以使代价最小，开摊代价为 $2 + 8 + 21 = 31$ 。