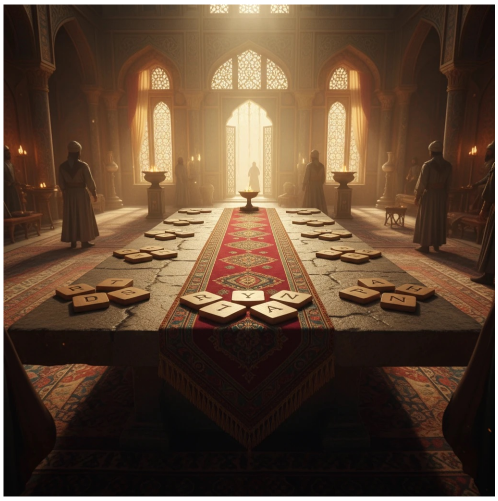
Problem A: Rayan Banner



The closing ceremony of the Rayan Programming Contest is underway, and you are tasked with creating a giant congratulatory banner for the winners. The banner is a long strip of fabric with movable letter tiles attached.

You are given a string s as the banner's initial text. You can rearrange the letters of s in any order. Your goal is to rearrange the letters to maximize the number of times the word *"rayan"* appears as a contiguous substring on the final arrangement.

Input

A single line containing a string s of length n ($1 \leq n \leq 1000$), consisting of lowercase English letters.

Output

Print a single integer, the maximum number of times the word *"rayan"* can appear after rearranging the letters of s.

Example

Standard Input	Standard Output
`yarane`	`1`
`ryane`	`0`

Problem B: Royal Persian Silk

In ancient times, the silk trade was prosperous, but it faced unique challenges. Merchants often struggled to sell fabrics containing minor flaws, such as small knots within the silk.

A silk fabric can be modeled as a rectangle, and each knot as a point on this rectangle. A piece of fabric is considered *flawless* if it doesn't contain any knots.

To deal with a flawed silk, a merchant can make exactly one straight cut—either horizontal or vertical—across the entire fabric. This cut splits it into two smaller rectangular pieces and removes any knot that it goes through. The value of each resulting piece equals its area **only if** it is flawless; otherwise, it is 0.

Your task is to determine the maximum possible total value of the two pieces after making the optimal cut.

Input

The first line contains three integers n ($1 \leq n \leq 1000$), W and H ($2 \leq W, H \leq 1000$), representing the number of knots, the width and height of the fabric, respectively.

Each of the next n lines contains two integers x and y ($0 < x < W$ and $0 < y < H$), representing the coordinates of a knot. All knot locations are distinct.

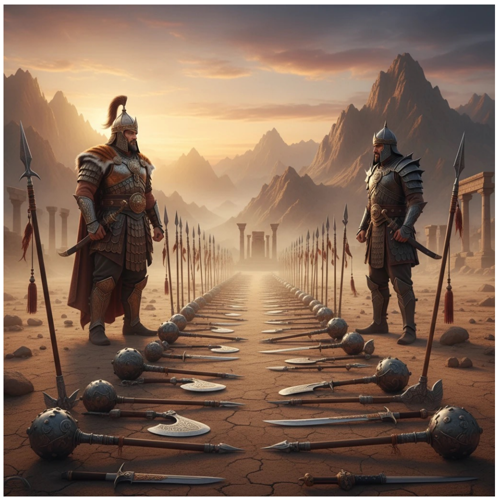
Output

Print a single integer, the maximum possible total value achievable after making one cut.

Example

Standard Input	Standard Output
4 3 5 1 1 2 2 1 3 2 4	5
1 1000 1000 500 500	1000000

Problem C: Weapon Ritual



After the tragic death of Siavash, the mighty Rostam faces the Turanian champion Ashkboos in battle. While Ashkboos has already prepared his weapon, Rostam demonstrates his confidence by inviting Ashkboos to a ritual to select his (Rostam's) weapon. The ritual proceeds as follows.

Initially, there is a row of n weapons, numbered from 1 to n from left to right, where the strength of the i^{th} weapon is p_i. There is also an empty sack.

Rostam goes first, and they alternate turns. In each turn, the current warrior does as follows:

- *Chooses* a weapon from the remaining weapons and adds it to the sack.
- Proceeds to either removes all the weapons to the left or all the weapons to the right of the chosen weapon from the row.

The ritual continues until no weapons remain in the row.

With k weapons collected in the sack, Rostam sorts them from weakest to strongest and selects the weapon at position $\lceil \frac{k+1}{2} \rceil$ (the median, rounded up) as his battle weapon.

Rostam aims to maximize the strength of his battle weapon, while Ashkboos aims to minimize it.

Assuming both warriors play optimally, what will the strength of Rostam's battle weapon be?

Input

The first line contains a single integer t ($1 \leq t \leq 10^4$), the number of test cases.

Each test case consists of two lines:

- The first line contains an integer n ($1 \leq n \leq 10^4$), the number of weapons.
- The second line contains n integers $p_1, p_2, \ldots, p_n$ ($1 \leq p_i \leq 10^9$), representing the strength of the weapons.

It is guaranteed that the sum of n over all test cases does not exceed 10^4.

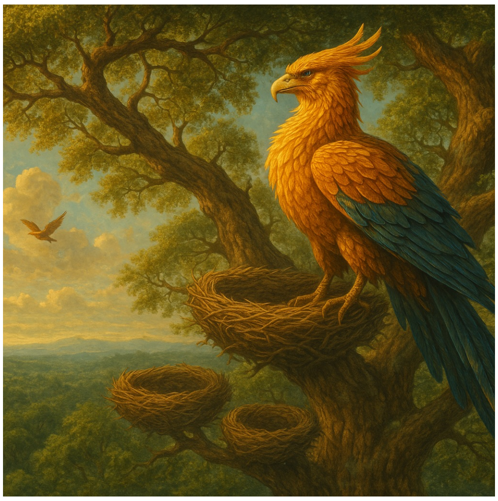
Output

Print a single integer, the strength of Rostam's battle weapon.

Example

Standard Input	Standard Output
2 5 1 2 3 4 5 5 1 1 1 1 1	5 1

Problem D: Simurgh's Nest



The benevolent Simurgh—a giant mythical bird—nests atop the Tree of Life, helping heroes like Zal and Rostam. With Nowruz approaching, she plans to build **at most** three nests on the vast tree, watching over the land.

The tree has n junctions (vertices) connected by branches (weighted edges). The nests must be placed on distinct vertices, with the constraint that the distance between any two nests must not exceed D.

The distance between two vertices u and v is the sum of the weights along the unique path connecting them.

Simurgh's goal is to minimize the maximum distance from any vertex to the nearest nest. Help Simurgh find this minimized value, and earn her magical feathers!

Input

The first line contains two integers n and D ($1 \leq n \leq 10^5$, $1 \leq D \leq 10^{14}$), the number of vertices and the maximum allowed distance between nests, respectively.

Each of the next $n - 1$ lines contains three integers u, v, w ($1 \leq u, v \leq n$, $1 \leq w \leq 10^9$), denoting an edge between u and v with weight w.

It is guaranteed that the input forms a tree.

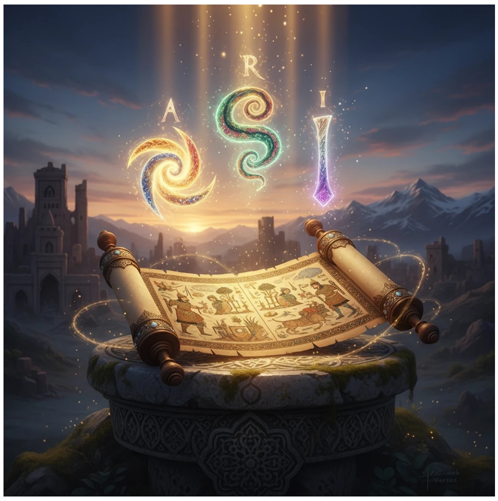
Output

Print a single integer, the minimum possible maximum distance from any vertex to its closest nest, with at most three nests on the tree.

Example

Standard Input	Standard Output
6 3 3 2 3 6 5 1 1 2 3 5 4 4 4 2 1	5
10 10 3 2 6 9 6 8 1 2 7 5 1 4 10 9 2 5 7 7 6 2 5 4 2 5 7 8 9	20
10 100 3 2 6 9 6 8 1 2 7 5 1 4 10 9 2 5 7 7 6 2 5 4 2 5 7 8 9	9

Problem E: Scroll of Aria



Tahmineh—the fierce princess who wed Rostam—dreams of her son Sohrab becoming a dragon slaying warrior. To foresee the future of Sohrab, she must learn the secrets of Arian history, hidden in the scrolls of Aria.

Each scroll contains a string s with even length, which consists of the letters A, R and I.

In order to decode a scroll, she must split s into disjoint pairs of letters such that:

- Each letter of s belongs to exactly one pair.
- Each pair, reading from left to right, is a substring of *ARIA* (i.e., *"AR"*, *"RI"*, or *"IA"*).

Letters in a pair do not necessarily need to be adjacent in string s.

For each scroll, Tahmineh wants to know whether it is possible to pair all of its letters according to these rules.

Input

The first line contains an integer t ($1 \le t \le 10^4$), representing the number of scrolls.

Then t lines follow, each containing a non-empty string s of even length, consisting only of letters A, R and I.

It is guaranteed that the total length of all strings does not exceed $2 \cdot 10^5$.

Output

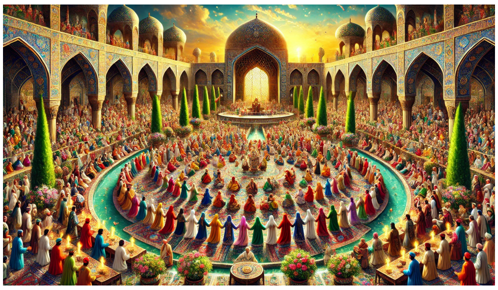
For each scroll, print *"Yes"* if it is possible to decode the scroll, or *"No"* otherwise.

The output is not case-sensitive.

Example

Standard Input	Standard Output
5 AR RIIR AIAR ARIA AIRRIA	Yes No Yes Yes Yes

Problem F: The Eternal Celebration



Once upon a time, during the annual celebration of Amordadegan—a festival symbolizing immortality and honoring the guardian of the world—people would gather and form circles. In each circle, every person holds hands with the person to their left and to their right. In case of a circle with only one person, they clasped their own hands.

Let there be n people labeled from 1 to n. In year 1, the initial hand-holding pattern consists of k circles.

The right-partner of a person is the person to their right.
Also, The right-partner of a person in a circle of length 1 is themself. Each year, the right-partner of a person will be the right-partner of their last year's right-partner.

Formally, let $R_{i,j}$ denote the person to the right of person i in year j. For each year $j > 1$, $R_{i,j} = R_{R_{i,j-1},j-1}$.

We say that the celebration remains novel up to year y if, for all $1 \leq i < j \leq y$, the hand-holding patterns of year i and year j are different. Two hand-holding patterns are considered the same if and only if for every person x, we have $R_{x,i} = R_{x,j}$.

Let Y be the last year for which the celebration is still novel. Given the lengths of the initial circles of year 1, find out the value of Y modulo $10^9 + 7$.

Input

The first line contains a single integer t ($1 \leq t \leq 100$) denoting the number of test cases.

The first line of each test case contains two integers n and k ($1 \leq n \leq 10^{14}$, $1 \leq k \leq 100$), denoting the total number of people and the number of circles in the initial pattern.

The second line of each test case contains k integers $a_1, a_2, \ldots, a_k$ ($1 \leq a_i \leq n$), representing the lengths of the circles in year 1. It is guaranteed that the sum of all a_i equals n.

It is guaranteed that the sum of n across all test cases is at most 10^{14}, and the sum of k across all test cases is at most 100.

Output

For each test case, print a single integer, the value of Y modulo $10^9 + 7$.

Example

Standard Input	Standard Output
3 10 1 10 7 3 3 3 1 7037035902234 3 1234567891011 2345678910111 ↩ ↪ 3456789101112	5 2 311076699

Problem G: The Great Vault of Secrets

Shaykh Bahā'ī, a brilliant 16th-century Persian scholar, invented remarkable creations including Sangak bread, a bathhouse heated by a single candle, and ingenious water distribution systems. Legend says he concealed all his secrets in a vault that can only be opened by those worthy enough to solve its puzzle.

The vault has n dials, where each dial i must be set to a value a_i between l_i and r_i (inclusive). XOR locks link some dials; for each XOR lock (x, y, z), the dial values must satisfy $a_x \oplus a_y = z$.

Your task is to find the lexicographically smallest sequence $a_1, a_2, \cdots, a_n$ that satisfies all range constraints and XOR locks. If no such sequence exists, output -1.

A sequence a is lexicographically smaller than a sequence b of the same length, if and only if, in the first position where a and b differ, the sequence a has a smaller number than the sequence b.

Input

The first line contains two integers n and m $(1 \le n \le 2 \cdot 10^5, 0 \le m \le 5 \cdot 10^5)$, the number of dials and the number of XOR locks.

The i^{th} line of the next n lines contains two integers l_i and r_i $(0 \le l_i \le r_i < 2^{30})$, the range for dial i.

Each of the next m lines contains three integers x, y, z $(1 \le x, y \le n, 0 \le z < 2^{30})$, describing an XOR lock.

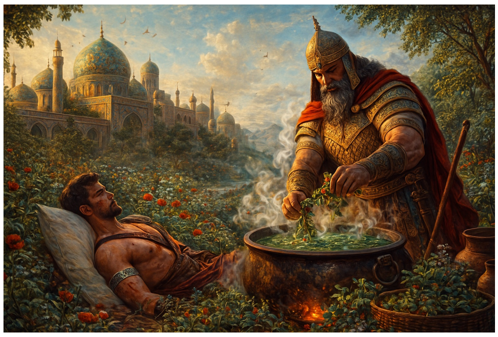
Output

If a valid sequence exists, output n space-separated integers $a_1, a_2, \cdots, a_n$, representing the lexicographically smallest solution; otherwise, output -1.

Example

Standard Input	Standard Output
5 3 1 3 1 4 1 4 2 7 4 6 1 2 3 3 4 1 4 5 7	1 2 2 3 4
5 3 1 3 1 4 1 4 4 7 4 6 1 2 3 3 4 1 4 5 7	-1
5 3 1 3 1 4 1 4 2 7 4 6 1 2 3 3 4 1 4 5 4	1 2 3 2 6

Problem H: The Elixir of Life



In the Shahnameh, Rostam mortally wounds an unknown young warrior, only to discover later that it is his own son, Sohrab. In despair, he seeks to brew a healing elixir using poppy flowers. There are n poppy gardens connected by $n-1$ roads, forming a tree, rooted at garden 1. Garden i contains a_i poppy flowers.

In the following days, Rostam attempts to gather as many poppy flowers as possible by harvesting the gardens. By harvesting garden i, Rostam collects a_i poppy flowers.

The owners of the gardens will only allow Rostam to harvest the gardens on one condition; Rostam needs to give all the harvested flowers on **even** days to the owners (day 2, 4, 6, etc). i.e. Rostam is only allowed to keep the harvest of **odd** days to himself (day 1, 3, 5, etc).

On day 1, Rostam harvests the root, collecting a_1 poppy flowers.

On day i ($i > 1$), as long as an unharvested garden remains, the harvesting proceeds as follows:

- Rostam defines *frontier* gardens: The set of all already-harvested gardens that have at least one unharvested child.
- For each frontier garden, Rostam will choose exactly one of its unharvested children gardens, and harvests it.

Note that newly harvested gardens may become frontier gardens the next day, if they have children.

Help Rostam plan his harvesting optimally, and keep as many poppy flowers as possible.

There are q updates, each changing the value of a_i for a single i. After each query, you should find the maximum amount of poppy flowers Rostam can collect for Sohrab. Updates are persistent, i.e., each change to the values is applied to all subsequent queries. Note that all the harvests are hypothetical, and the only change to the amount of flowers in gardens comes from the updates.

Input

The first line contains two integers n and q ($1 \le n \le 5 \cdot 10^5, 1 \le q \le 3 \cdot 10^5$), the number of gardens and the number of updates.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($0 \le a_i \le 10^9$), initial flower amounts in gardens 1 to n.

The next $n-1$ lines contain two integers u and v ($1 \le u, v \le n$), describing an undirected edge connecting garden u and garden v.

The following q lines each contain two integers i and x ($1 \le i \le n$, $0 \le x \le 10^9$), meaning the flower amount in garden i is updated to x.

It is guaranteed that the edges form a tree.

Output

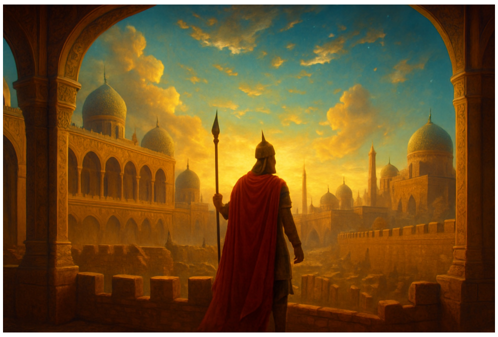
Print $q+1$ lines:

- The first line: Rostam's maximum amount of flowers, with the initial flower amounts in gardens.
- The next q lines: Rostam's maximum amount of flowers, immediately after each update, as if harvesting started from scratch with the new amounts.

Example

Standard Input	Standard Output
3 5 1 1 1 1 2 2 3 1 5 2 5 3 5 1 4 2 10	2 6 6 10 9 9

Problem I: Frontier Watch



Fereydun is a heroic king who rises against the tyrant Zahhak to free the people from oppression. Before his uprising, Fereydun stationed scouts along the border at milestones spaced one unit apart. Each scout has a role from 1 to k, such as rider, archer, and messenger.

Scouts may arrive or leave over time. After each change, Fereydun wants to know the length of the shortest continuous segment of the border that contains at least one scout of every role. The length of a segment is defined as the difference between its maximum and minimum positions.

Input

The first line contains two integers n and k ($1 \leq k \leq n \leq 2 \cdot 10^5$), the number of initial scouts and the number of roles.

Each of the next n lines contains two integers x_i and t_i ($1 \leq x_i \leq 10^9$, $1 \leq t_i \leq k$), the position and role of an initial scout.

The next line contains an integer q ($1 \leq q \leq 2 \cdot 10^5$), the number of queries.

Each of the next q lines is one query of the following form:

- $+\ x\ t$: add a scout of role t at position $\text{dec}(x)$ ($0 \leq x \leq 10^9 + 6$, $1 \leq t \leq k$),
- $-\ x$: remove the scout at position $\text{dec}(x)$ ($0 \leq x \leq 10^9 + 6$).

The positions x in queries are encoded. Let ans_i denote the answer to the i-th query, and let ans_0 be the answer for the initial set of scouts (i.e., the length of the shortest segment containing at least one scout of every role before any queries). The real position for the i-th query is computed as:

$$\text{dec}(x_i) = (x_i \cdot \text{ans}_{i-1}) \bmod (10^9 + 7).$$

It is guaranteed that:

- All positions are distinct initially.
- Throughout the queries, there is never more than one scout at the same position.
- At every moment, there is at least one scout of each role.
- For removal queries, a scout exists at position $\text{dec}(x)$.
- For all queries, $\text{dec}(x_i)$ is in the range $[1, 10^9]$.

Output

After each query, print a single integer, the length of the shortest segment containing at least one scout of every role.

Example

Standard Input	Standard Output
4 3 1 3 7 6 1 2 2 3 5 + 1 3 + 1 1 + 1 1 - 500000004 - 2	4 2 2 2 3
7 7 1 2 3 4 7 9 11 2 6 7 4 3 5 1 7 + 1 3 + 200000002 5 - 200000002 - 1 + 200000002 2 + 555555560 7 + 111111113 1	10 10 10 10 9 9 8