

Problem A. Clan Battle

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

Hikawa Kyouka is participating in her clan battle, where she needs to defeat m types of bosses. Each boss can appear in multiple rounds. All bosses start in round 1 and are initially available to challenge. For a boss i to become available in round j ($j \geq 2$), **both** conditions must be met:

- The same boss i of the immediately preceding round ($j - 1$) is defeated.
- All bosses of the earlier round ($j - k$) have been defeated.

For example, in the picture below, because the third boss is of Round 60, although the first boss of Round 61 is defeated, that boss of Round 62 won't appear.



An example of clan battle when $k = 2$. The *defeated* label suggests that the first, fourth, and fifth bosses are unavailable.

Kyouka wants to defeat n bosses in order $a_1, a_2, \dots, a_n$. Can you help her determine if it is possible?

Input

Each test contains multiple test cases. The first line contains one integer t ($1 \leq t \leq 10^4$) — the number of test cases. The description of each test case follows.

The first line of each test case contains three integers n , m , and k ($1 \leq n \leq 4 \cdot 10^6$, $1 \leq m < 10^6$, $1 \leq k \leq 10^9$).

The second line of each test case contains n integers $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq m$).

It is guaranteed that the sum of n over all test cases doesn't exceed $4 \cdot 10^6$.

Output

If Kyouka can defeat the bosses in the order, output “YES”; otherwise, output “NO”. You can output the answer in any case (upper or lower). For example, the strings “yEs”, “yes”, “Yes”, and “YES” will be recognized as positive responses.

Example

standard input	standard output
4	YES
5 2 2	YES
1 1 2 2 1	NO
5 2 2	YES
2 1 1 1 2	
5 3 2	
2 1 1 1 2	
10 4 3	
4 3 3 2 2 1 1 1 1 4	

Problem B. Nabi Puzzle

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

Izumi Yuki retaliates against Kayamori Ruka’s constant use of the verbal quirk *teheperinko* by creating a puzzle. Ruka finds herself at Nabi’s ground where n Nabis are arranged in a straight line. Yuki’s AI assistant Ketsu reveals the challenge: assign integer values to the Nabis such that:

- Every consecutive group of p Nabis (moving left-to-right) has a positive total sum.
- Every consecutive group of q Nabis has a negative total sum.

Help Ruka solve the puzzle!

Input

Each test contains multiple tests. The first line contains one integer t ($1 \leq t \leq 10^4$) — the number of test cases. The description of each test case follows.

The only line of each test case contains three integers p , q , and n ($1 \leq p, q, n \leq 2 \cdot 10^5$).

It is guaranteed that the sum of n over all test cases doesn’t exceed $2 \cdot 10^5$.

Output

For each test case:

- If a solution exists, output “YES” in the first line, and then output n integers $a_1, a_2, \dots, a_n$ ($|a_i| \leq 10^{12}$) in the second line, where a_i indicates the integer assigned to the i -th Nabi. It can be proven that if such a sequence exists, there will always be one that satisfies the given constraints.
- If a solution doesn’t exist, output “NO” in the single line.

You can output the answer in any case (upper or lower). For example, the strings “yEs”, “yes”, “Yes”, and “YES” will be recognized as positive responses.

Example

standard input	standard output
4	YES
2 3 3	-2 3 -2
2 3 4	NO
6 6 5	YES
7 11 14	1 1 1 1 1
	YES
	1 1 5 -13 1 1 7 1 1 3 -11 1 1 7

Problem C. Dominance

Input file: **standard input**
Output file: **standard output**
Time limit: 5 seconds
Memory limit: 1024 megabytes

To help Noumi Kudryavka study math, you, as the leader of Little Busters, come up with a game on matrices. The initial matrix has n rows and m columns and contains $n \cdot m$ **distinct** integers from 1 to $n \cdot m$.

- The i -th row is dominant if and only if for all $1 \leq j \leq m$, $a_{i,j}$ is the **minimum** value of the j -th column.
- The j -th column is dominant if and only if for all $1 \leq i \leq n$, $a_{i,j}$ is the **maximum** value of the i -th row.

If a row or column is dominant, Kudryavka can remove the row or column, and the remaining elements keep their order. If Kudryavka can remove the entire matrix by this operation, then you consider the matrix to be *good*.

You're lazy to generate such matrices, and you assign the work to others. However, you must write a checker to confirm the matrices are *good*.

Input

Each test contains multiple test cases. The first line contains one integer t ($1 \leq t \leq 10^4$) — the number of test cases. The description of each test case follows.

The first line of each test case contains two integers n and m ($1 \leq n, m \leq 10^6$, $1 \leq n \cdot m \leq 10^6$) — the number of the rows and columns in the matrix.

The i -th line of the next n lines of each test case contains m integers $a_{i,1}, a_{i,2}, \dots, a_{i,m}$ ($1 \leq a_{i,j} \leq n \cdot m$). It is guaranteed that $a_{i,j}$ are pairwise distinct.

It is guaranteed that the sum of $n \cdot m$ over all test cases doesn't exceed 10^6 .

Output

For each test case, output “YES” if the matrix is *good*; otherwise, output “NO”. You can output the answer in any case (upper or lower). For example, the strings “yEs”, “yes”, “Yes”, and “YES” will be recognized as positive responses.

Example

standard input	standard output
2	YES
2 2	NO
1 2	
3 4	
2 2	
1 4	
3 2	

Problem D. Greedy Counting

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

Shimoe Koharu is given an array a and required to find the length of its longest increasing sequence. Since Koharu doesn't know the classical problem nor the dynamic programming approach, she picks the sequence by greedy:

1. Let $i := 1$ be the current index and the answer be 1.
2. For $j > i$, pick the smallest j such that $a_j > a_i$. If she cannot find such j , end the procedure.
3. Let $i := j$ and increase the answer by 1. Then jump to Step 2.

This algorithm is clearly wrong: for example, if $a = [1, 4, 2, 3]$, she will get $[1, 4]$ instead of the correct $[1, 2, 3]$. As the consultant of the Supplemental Lessons Club, you're a bit annoyed and assign her to find such sequences on all subarrays of a via her approach, before she needs to sum up the length of these sequences.

However, you need to judge if her result is right. So it's now your task to calculate the sum.

Input

Each test has multiple test cases. The first line contains one integer t ($1 \leq t \leq 10^4$) — the number of test cases. The description of each test case follows.

The first line of each test case contains one integer n ($1 \leq n \leq 4 \cdot 10^5$).

The second line of each test case contains n integers $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq n$).

It is guaranteed that the sum of n over all test cases doesn't exceed $4 \cdot 10^5$.

Output

For each test case, output one integer — the sum of the length of all sequences obtained by Koharu's approach on all subarrays of a .

Example

standard input	standard output
3	1
1	14
1	39
4	
1 4 2 3	
6	
1 1 4 5 1 4	

Problem E. Ever Forever

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

Hayama Mizuki is learning English and discovers an interesting linguistic pattern: the word “**ever**” appears as a suffix in “**forever**”. Inspired by this, she creates a dynamic vocabulary system and poses a challenge to you.

Formally, she maintains a vocabulary list that undergoes the following operations:

- Words can be added to or removed from the list. It is guaranteed that there are no duplicates in the vocabulary.
- After modifications, you must determine the number of ordered pairs (A, B) where:
 - A is a suffix* of B .
 - A and B are distinct and both exist in the current vocabulary.

Input

The first line contains one integer n ($1 \leq n \leq 10^5$) — the number of operations that Mizuki does.

Each of the following n lines contains n operations. Each operation is one of the two types:

- “+ s ” — Mizuki adds word s that contains only lower case Latin characters to the vocabulary. It is guaranteed that s doesn’t exist in the vocabulary before the operation.
- “– s ” — Mizuki removes word s from the vocabulary. It is guaranteed that s exists in the vocabulary before the operation.

Note that a removed word can be readded into the vocabulary.

It is guaranteed that the sum of $|s|$ over all operations doesn’t exceed 10^6 .

Output

Output q integers. Each integer indicates the answer after Mizuki’s each operation.

Example

standard input	standard output
10	0 1 2 5 3 6 8 9 4 4
+ ever	
+ never	
+ forever	
+ er	
– never	
+ rever	
+ never	
+ father	
– er	
+ mother	

*A string a is a suffix of a string b if a can be obtained from b by deletion of several (possibly, zero or all) characters from the beginning. For example, “**ever**” is a suffix of “**forever**”, but “**abc**” is not a suffix of “**aabbc**”

Problem F. Pull

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

In the game *Arknights*, you can cost materials to headhunt an operator, which is known as a *pull*. Initially, the 6-star operators' rate is $a\%$. In other words, the probability that you obtain a 6-star operator is $a\%$. If a 6-star operator does not appear after c pulls, each subsequent pull will increase the 6-star operators' rate by $b\%$, up to 100%. For example, on the $(c + 1)$ -th pull, the rate increases to $(a + b)\%$, and then on the $(c + 2)$ -th pull, the rate increases to $(a + 2b)\%$. The pull count and increased rate will be reset as soon as a 6-star operator appears on any pull.

Given a , b , and c , find the expected pulls between two 6-star operators. In other words, if you have just obtained a 6-star operator, find the expected pulls that you need to obtain a next 6-star operator.

Input

Each test contains multiple tests. The first line contains one integer t ($1 \leq t \leq 10^3$) — the number of test cases. The description of each test case follows.

The only line of each test case contains three integers a , b , and c ($1 \leq a, b, c \leq 100$).

Output

For each test case, output one real number — the expected pulls between two consecutive 6-star operators.

Your answer is considered correct if its absolute or relative error does not exceed 10^{-6} . Formally, let your answer be a , and the jury's answer be b . Your answer is accepted if and only if $\frac{|a-b|}{\max(1,|b|)} \leq 10^{-6}$.

Example

standard input	standard output
3	12.20996063021597
1 1 1	34.59455493520978
2 2 50	1.000000000000000
100 100 100	

Problem G. DSU

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

Since her creation to maintain the Tethys system, the Shorekeeper has diligently studied various algorithms under her creator's guidance. Among these, she found the Disjoint Set Union (DSU) data structure to be the most elegant. Her implementation uses the following code:

```
vector<int> f(n); iota(f.begin(), f.end(), 0); // initialize f[i] = i
int find(int u){ return f[u] == u ? u : f[u] = find(f[u]); } // path compression
void merge(int u, int v){ u = find(u); v = find(v); f[u] = v; }
```

Your task is to determine whether a given array f could result from applying at most n **merge** operations on an initially created DSU structure. The initial state uses the first code snippet above, and subsequent operations may only call the **merge** function.

Input

Each test has multiple test cases. The first line contains one integer t ($1 \leq t \leq 10^4$) — the number of test cases. The description of each test case follows.

The first line of each test case contains one integer n ($1 \leq n \leq 4 \cdot 10^5$).

The second line of each test case contains n integers $f_0, f_1, \dots, f_{n-1}$ ($0 \leq f_i < n$).

It is guaranteed that the sum of n over all test cases doesn't exceed $4 \cdot 10^5$.

Output

For each test case, if a solution exists, output as follows:

- The first line contains one integer k ($0 \leq k \leq n$) — the number of **merge** calls.
- The following k lines each contain two integers u and v ($0 \leq u, v < n$), indicating calling **merge**(u, v).

; otherwise, output -1 in a single line.

Example

standard input	standard output
3	0
1	-1
0	1
2	2 0
1 0	
3	
0 1 0	

Problem H. Simple XOR Problem

Input file: `standard input`
Output file: `standard output`
Time limit: 2.5 seconds
Memory limit: 256 megabytes

Since Nagasaki Soyo claimed that she would do anything, she was given the simple XOR problem. Help her!

You're given four integers l , r , y , and k . Calculate:

$$\sum_{x=l}^r (x \oplus y)^k \pmod{(10^9 + 7)}$$

where $\oplus$ denotes the bitwise XOR operation.

Input

The only line contains four integers l , r , y , and k ($1 \leq l \leq r \leq 10^9$, $1 \leq y \leq 10^9$, $1 \leq k \leq 10^6$).

Output

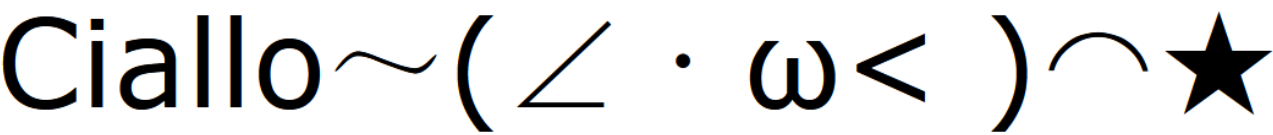
Output one integer indicating the result of the formula above.

Examples

standard input	standard output
1 10 1 1	55
114 514 1919 810	353713127

Problem I. Ciallo

Input file: standard input
Output file: standard output
Time limit: 4 seconds
Memory limit: 1024 megabytes



Inaba Meguru, the creator of the viral *Ciallo*, seeks to expand her linguistic creations by combining prefixes and suffixes from two base words. Formally, she has two strings s and t , and she will create new strings by concatenating a prefix[†] of s and a suffix[‡] of t . She wants to know how many different non-empty strings she can obtain in this way.

Input

Each test contains multiple tests. The first line contains one integer T ($1 \leq T \leq 10^4$) — the number of test cases. The description of each test case follows.

The first line of each test contains a string s ($1 \leq |s| \leq 10^6$).

The second line of each test contains a string t ($1 \leq |t| \leq 10^6$).

It is guaranteed that both s and t contain only lower case Latin characters.

It is guaranteed that neither the sum of $|s|$ nor the sum of $|t|$ over all test cases exceeds 10^6 .

Output

For each test case, output an integer indicating the number of different strings Meguru can obtain.

Example

standard input	standard output
4	2
a	3
a	28
a	122
b	
ciao	
hello	
inabameguru	
ayachinene	

[†]A string a is a prefix of a string b if a can be obtained from b by deletion of several (possibly, zero or all) characters from the end. For example, “for” is a prefix of “forever”, but “abc” is not a prefix of “aabbcc”

[‡]A string a is a suffix of a string b if a can be obtained from b by deletion of several (possibly, zero or all) characters from the beginning. For example, “ever” is a suffix of “forever”, but “abc” is not a suffix of “aabbcc”

Problem J. Bridge III

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

Bidding is the first phase of a bridge game where players communicate their hand strength and distribution to determine the contract (the final goal for the round). The rules of bidding are listed as follows.

- **Bid Structure:**

- An auction consists of a number (from 1 to 7) and a suit.
- Suit hierarchy (lowest to highest): Club (C), Diamond (D), Heart (H), Spade (S), No Trump (N).
- To **outrank** another auction:
 - * Higher number, **OR**
 - * Same number but higher-ranked suit.
 - * Example: 2C > 1S (higher number), 2N > 2C (higher suit).

- **Players and Turns:**

- Players take turns in order: $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 1 \dots$
- Partnerships: Player 1 & 3 are partners; Player 2 & 4 are partners.
- Opponents are non-partner players.

- **Allowed Actions:**

- **Auction:** Bid higher than the last valid auction.
 - * If there is no previous valid auction, the player can bid any auction.
- **Pass (P):** Skip without bidding.
- **Double (X):**
 - * Only allowed if the last non-pass bid was an *auction* by an opponent.
- **Redouble (XX):**
 - * Only allowed if the last non-pass bid was a *double* by an opponent.

- **Ending Conditions:**

- All four players pass initially, **OR**
- Three consecutive passes occur after any non-pass bid.

You're given a bidding sequence. Determine if it is both **valid** (follows all rules) and **complete** (meets an ending condition).

Input

Each test contains multiple tests. The first line contains one integer t ($1 \leq t \leq 100$) — the number of test cases. The description of each test case follows.

The first line of each test case contains one integer n ($1 \leq n \leq 320$) — the length of the bidding sequence.

The second line contains n strings which indicate each player's bids. It is guaranteed that the bids are valid.

Output

For each test case, if the bidding sequence is valid, output “YES”; otherwise, output “NO”. You can output the answer in any case (upper or lower). For example, the strings “yEs”, “yes”, “Yes”, and “YES” will be recognized as positive responses.

Example

standard input	standard output
6	YES
4	NO
P P P P	YES
5	NO
P P P P X	NO
5	YES
P 1C P P P	
6	
P P 1C 1D P P	
8	
P 1H X XX X P P P	
12	
1S P 2C P 2D P 4S X XX P P P	

Problem K. Painting the Tree

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

Although Ororon is from the *Master of the Night Wind*, his weaving skills leave much to be desired. His creations resemble patterned towels more than proper woven scrolls. When the tribe's master artisans abandon hope of improving his craftsmanship, Ororon turns to sketching designs with pen and paper.

His grandmother Citlali, the renowned *Granny Itztli*, devises a challenge. She provides a tree with n vertices and m colored path operations. The operations are divided into two types:

For the operation of the first type, Ororon is given a vertex pair (u, v) and a color c . He must paint all vertices along the unique simple path between u and v with color c . Each painting operation completely overwrites previous colors on those vertices.

For the operation of the second type, Ororon is also given a vertex pair (u, v) and a color c . He must paint all vertices of the subtree[§] of u with c , assuming v is the root of the tree. Note that v is assumed as the root **only in the current operation**.

Being an expert shaman, Citlali already knows all final colors — her goal is to verify Ororon's work. Of course, Ororon also hates tedious painting, so help him determine the final color of every vertex after processing all m operations in order.

Input

Each test has multiple test cases. The first line contains one integer t ($1 \leq t \leq 10^4$) — the number of test cases. The description of each test case follows.

The first line of each test case contains two integers n and m ($1 \leq n \leq 4 \cdot 10^5$, $1 \leq m \leq 2 \cdot 10^5$).

The second line of each test case contains n integers $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 10^6$) — the initial color of each vertex.

The i -th line of the following $n - 1$ lines contains two integers x_i and y_i ($1 \leq x_i, y_i \leq n$), describing an edge between vertex x_i and y_i . It is guaranteed that these edges form a tree.

The i -th line of the following m lines contains four integers op_i , u_i , v_i and c_i ($1 \leq op_i \leq 2$, $1 \leq u_i, v_i \leq n$, $1 \leq c_i \leq 10^6$) — the type of the i -th operation and its parameters.

It is guaranteed that the sum of n over all test cases doesn't exceed $4 \cdot 10^5$, and the sum of m over all test cases doesn't exceed $2 \cdot 10^5$.

Output

For each test case, output n integers — the final color of each vertex.

[§]The subtree of a vertex u is the set of all vertices that pass through u on a simple path to the root.

Example

standard input	standard output
2	5
1 2	1 1 1 6 1 6 2 1
1	
1 1 1 4	
2 1 1 5	
8 4	
1 2 3 4 5 6 7 8	
8 1	
6 7	
4 6	
7 5	
3 2	
5 2	
5 1	
2 4 4 5	
1 3 4 6	
1 1 7 2	
2 5 6 1	

Problem L. Fireflies

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

This is an interactive problem.

You're invited by Firefly to a garden, where 10^5 trees are lined up. The trees are numbered from 1 to 10^5 from left to right. She tells you she has put n groups of fireflies, each group located at a distinct secret tree x_i , and she would like to play a game with you. You can query a coordinate x , and Firefly will tell you the sum of the Manhattan distance[¶] between your guess and each group's coordinates.

However, there is little time for Firefly to spend with you, and you have to **come up with one of the groups' coordinates within 40 queries**. If you manage to do so, she can take you to the coordinate and enjoy the fireflies with you. If you fail, she will have to say farewell and carry out her mission.

Input

Each test contains multiple tests. The first line contains one integer t ($1 \leq t \leq 10^3$) — the number of test cases. The description of each test case follows.

The first line contains one integer n ($1 \leq n \leq 10^4$) — the number of firefly groups.

It is guaranteed that the sum of n over all test cases doesn't exceed 10^4 .

Interaction Protocol

To make a query, output a line in the following format (do not include quotes):

"? x " ($1 \leq x \leq 10^5$)

Here, x represents your query coordinate.

After each query, you should read a line containing one integer, indicating the sum of the Manhattan distance between your guess and each group's coordinates. In other words, the integer is $\sum_{i=1}^n |x_i - x|$.

When you are ready to print the answer, output a single line in the following format:

"! x " ($1 \leq x \leq 10^5$)

Here, x represents your answer coordinate.

You can make at most 40 queries in each test case.

The interactor is NOT adaptive, meaning that the answer is known before the participant asks the queries and doesn't depend on the queries asked by the participant.

If your program makes more than 40 queries for one test case, your program should immediately terminate to receive the verdict **Wrong Answer**. Otherwise, you can get an arbitrary verdict because your solution will continue to read from a closed stream.

After printing a query do not forget to output the end of line and flush the output. Otherwise, you may get **Idleness limit exceeded** verdict. To do this, use:

- `fflush(stdout)` or `cout.flush()` in C++;
- `System.out.flush()` in Java;
- `Console.Out.Flush()` in C#;
- `stdout.flush()` in Python;

[¶]The Manhattan distance between two coordinates x_1 and x_2 is given by $|x_1 - x_2|$.

- see the documentation for other languages.

Example

standard input	standard output
2	
7	? 12
34	? 5
25	! 1
4	! 42657

Note

In the first sample case, $x = [13, 5, 10, 8, 9, 6, 1]$.

In the second sample case, $x = [85688, 42657, 37978, 85893]$.

Problem M. Magical Book

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

As the manager of the magical library of the Yuugyuuji's, Chrysoberyl can extract any contiguous subsequence from an existing magical book to create a new tome. The magical books are composed solely of parentheses "(" and ")". A book is considered *beautiful* if its content strictly adheres to the following formal grammar:

- The atomic unit $()$ is *beautiful*.
- For any *beautiful* book A , the wrapped form (A) is also *beautiful*.
- For any two *beautiful* books A and B , their concatenation AB preserves its *beauty*.

Chrysoberyl possesses q identical old magical books. Due to the ancient nature of these books, their opening and closing sections have become unusable. Specifically, for the i -th book, she **must** select a substring between positions l_i and r_i (inclusive). Two selection methods are considered different if and only if they have distinct starting positions **or** distinct ending positions, even if their content is identical.

Input

The first line contains two integers n and q ($1 \leq n, q \leq 2 \cdot 10^5$) — the length of each magical book's content and the number of the magical books.

The second line contains n characters $s_1, s_2, \dots, s_n$ ($s_i = '('$ or $)'$) — each magical book's content.

Each of the following q lines contains two integers l_i and r_i ($1 \leq l_i \leq r_i \leq n$).

Output

Output q integers. The i -th integer represents the number of different ways that Chrysoberyl can create a beautiful book from the i -th book.

Example

standard input	standard output
12 7 () (()) (() 1 1 2 3 1 2 1 12 8 12 5 11 2 10	0 0 1 6 2 3 3