The 2nd Universal Cup

Stage 28: Chengdu

April 13-14, 2024

This problem set should contain 13 problems on 20 numbered pages.

Based on

China Collegiate Programming Contest (CCPC)

Hosted by

Problem A. Add One 2

Input file:	standard input
Output file:	standard output
Time limit:	1 second
Memory limit:	1024 megabytes

Little Cyan Fish and his *once-good* friend released a problem called "Add One" in the summer of 2022[1]. His friend was very fascinated by the concept of adding one and came up with many ideas for it.

Two years have passed, and the ideas from back then have become vague. Now, Little Cyan Fish wants to play the following game with Little Leaky Fish.

In this game, Little Leaky Fish has a sequence of integers $x_1, x_2, \cdots, x_n$. Initially, all the values of x_i are set to 0. He can then execute the following operations any number of times:

- Choose an integer k ($1 \le k \le n$). Then, add one to each x_i for $1 \le i \le k$. The cost of such operation is k.
- Choose an integer k ($1 \le k \le n$). Then, add one to each x_i for $n - k + 1 \le i \le n$. The cost of such operation is k.

Little Cyan Fish also has n integers $y_1, y_2, \cdots, y_n$. He wishes for Little Leaky Fish to perform a series of operations such that the condition $x_i \ge y_i$ holds for all $1 \le i \le n$.

This task is too easy for Little Leaky Fish. So he is wondering what is the minimum cost required to meet the goal. Therefore, your task is to write a program to calculate the minimum cost.

Input

There are multiple test cases in a single test file. The first line of the input contains a single integer T ($1 \le T \le 10^5$), indicating the number of test cases.

For each test case, the first line of the input contains a single integer n ($1 \le n \le 10^6$).

The next line of the input contains n integers $y_1, y_2, \cdots, y_n$ ($0 \le y_i \le 10^9$).

It is guaranteed that the sum of n over all test cases does not exceed 10^6.

Output

For each test case, output a single line contains a single integer, indicating the answer.

Example

standard input	standard output
3 5 1 1 1 1 1 10 0 0 0 0 1000000000 0 0 0 0 0 13 1 1 4 5 1 4 1 9 1 9 8 1 0	5 5000000000 76

[1] Add One from Petrozavodsk Programming Camp, Summer 2022. https://qoj.ac/problem/4810

Problem B. Periodic Sequence

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

This is another story of Kevin, a friend of Little Cyan Fish.

Kevin is the chief judge of the International Convex Polygon Championship (ICPC). He proposed a geometry task for the contest. However, since he is inexperienced in computational geometry, he couldn't generate a correct convex polygon for the tests of the task. Thus, he shifted his focus to a string-related problem.

In this problem, we will assume all strings consist of lowercase letters only. For a string $S = S_0 S_1 \cdots S_{|S|-1}$, we will use $|S|$ to denote the length of the string, and S_i to denote the $(i+1)$-th character of the string. For instance, for $S = \texttt{xiaoqingyu}$, it holds that $|S| = 10$, with $S_0 = \texttt{x}$, $S_1 = \texttt{i}$, and $S_9 = \texttt{u}$.

A string T is defined as a *period* of another string S if and only if for every $0 \le i < |S|$, the equality $S_i = T_{i \bmod |T|}$ holds. For example, "`ccpc`" is a period of "`ccpcccpc`" and "`ccpccc`", whereas "`cpc`" is not a period of "`ccpc`".

Kevin defines that a sequence of strings $[S_1, S_2, \cdots, S_k]$ is called *periodic* if and only if it satisfies:

- $S_i \ne S_j$ for all $1 \le i < j \le k$
- S_i is a period of S_{i+1} for all $1 \le i < k$

Kevin loves the concept of *periodic*, so he asks Little Cyan Fish the following problem:

- For a given integer n, what is the length (denoted by ℓ) of the longest periodic sequence $S_1, S_2, \cdots, S_\ell$, satisfying $|S_i| \le n$ for all $1 \le i \le \ell$.

Let $f(n)$ be the answer to the problem above for a fixed integer n. Little Cyan Fish feels the problem is too easy, so he is wondering the value of $f(1), f(2), \ldots, f(N)$. Can you help him to calculate the values?

Since the values can be huge, you only need to output the answers modulo a given prime number M.

Input

The first line of the input contains two integers N and M ($1 \le N \le 2 \times 10^5$, $5 \times 10^8 \le M \le 1.01 \times 10^9$).

It is guaranteed that M is a prime number.

Output

Output a single line with N integers, indicating the values of $f(1), f(2), \ldots, f(N)$, modulo M.

Example

standard input	standard output
`5 1000000007`	`1 3 6 11 19`

Note

For the first testcase, we have $f(1) = 1$, $f(2) = 3$, $f(3) = 6$.

For $n = 1$, one of the possible periodic sequences is [`a`].

For $n = 2$, one of the possible periodic sequences is [`ab`, `a`, `aa`].

For $n = 3$, one of the possible periodic sequences is [`abc`, `ab`, `aba`, `a`, `aaa`, `aa`]

Problem C. Colorful Graph 2

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

Having delved into the complex theory of quantum chromodynamics again, Little Cyan Fish has become fascinated with the concept of color charge. To test your understanding of this theory, he has proposed the following task to you.

Consider a regular polygon (i. e., a polygon with all sides having the same length and all angles having the same value) with vertices numbered in clockwise order by integer numbers from 0 to $n-1$. Let its vertices correspond to vertices of an **undirected** graph. For each i ($0 \leq i < n$), there is an edge connecting the vertex i and $(i+1) \bmod n$. Additionally, there are m extra edges in the graph, where the i-th edge connects the vertex u_i and v_i. It is guaranteed that these additional m edges are pairwise distinct, and all the m edges are unique compared to the n edges forming the polygon and that none of the edges intersect at non-vertex points.

Little Cyan Fish would like you to color all the vertices into two colors: black and red. But Little Cyan Fish wants the graph to be colorful – each cycle in the graph must contain two kinds of colors. Formally, he does **not** want to have a sequence of vertices $v_0, v_1, \cdots, v_{t-1}$ ($t \geq 3$) satisfying:

- The color of $v_0, v_1, \cdots, v_{t-1}$ are the same (i.e. all the vertices are black/red).
- For each $0 \leq i < t$, there is an edge connecting the vertex v_i and $v_{(i+1) \bmod t}$.

Your task is to show him a possible coloring plan or report that there is no possible solution.

Input

There are multiple test cases in a single test file. The first line of the input contains a single integer T ($1 \leq T \leq 10^5$), indicating the number of test cases.

For each test case, the first line of the input contains two integers n and m ($3 \leq n \leq 2\times10^5$, $0 \leq m \leq n-3$), indicating the number of the vertices of the polygon, and the number of the extra edges.

The following m lines describes the extra edges. The i-th line of these lines contains two integers u_i and v_i ($0 \leq u_i, v_i \leq n-1$, $u_i \neq v_i$), indicating an extra edge. It is guaranteed that these additional m edges are pairwise distinct, and all the m edges are unique compared to the n edges forming the polygon and that none of the edges intersect at non-vertex points.

It is guaranteed that the sum of n over all test cases does not exceed 10^6.

Output

For each test case:

- If there is a possible plan, output a single line contains a single string of length n, indicating the plan. Each character of the string must be either "`B`" or "`R`", indicating the color of each vertex. If there are multiple possible solutions, you may print any of them.
- Otherwise, print a single line "`Impossible`".

Example

standard input	standard output
3 3 0 4 1 1 3 6 3 0 2 2 4 4 0	BRR BRBR RRBRRB

Problem D. Min or Max

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

Little Cyan Fish has recently embarked on a journey to understand the concepts of minimum and maximum values. Today, he has found a sequence of integers $a_1, a_2, \cdots, a_n$. He can perform the following two types of operations any times:

- Select two adjacent elements, x and y, in the sequence and merge them into a single element equal to $\min(x, y)$.
- Select two adjacent elements, x and y, in the sequence and merge them into a single element equal to $\max(x, y)$.

When merging two adjacent elements, these two elements will be removed from the sequence. The new element will be inserted at the original position of the two elements.

Little Cyan Fish wants to transform the original sequence $a_1, a_2, \cdots, a_n$ into another sequence $b_1, b_2, \cdots, b_m$. Your mission is to determine if it is possible to achieve this target sequence through a sequence of operations.

Input

There are multiple test cases in a single test file. The first line of the input contains a single integer T ($1 \le T \le 10^5$), indicating the number of test cases.

For each test case, the first line of the input contains two integers n and m ($1 \le m \le n \le 10^5$).

The next line of the input contains n integers $a_1, a_2, \cdots, a_n$ ($1 \le a_i \le n$).

The next line of the input contains m integers $b_1, b_2, \cdots, b_m$ ($1 \le b_i \le n$).

It is guaranteed that the sum of n over all test cases does not exceed 10^5.

Output

For each test case, if it is possible to transform the sequence, output a single line `Yes`. Otherwise, output a single line `No`.

Example

standard input	standard output
3 2 1 1 2 1 5 4 1 1 1 1 1 1 1 2 1 10 5 1 2 3 4 5 6 7 8 9 10 2 4 6 8 10	Yes No Yes

Problem E. Min or Max 2

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

The story continues. Now Little Cyan Fish has mastered the concept of Min or Max. He would like to practice his understanding by playing with some tuples of integers.

Here, Little Cyan Fish is presented with two sequences, each a permutation of the numbers from 1 to n, which we will call $a_1, a_2, \cdots, a_n$ and $b_1, b_2, \cdots, b_n$. Initially, he has a tuple (x, y), where $x = a_1$ and $y = b_1$. For each subsequent index i, ranging from 2 to n, he must choose and execute exactly one of the following operations in ascending order of i:

- Update $x \leftarrow \min(x, a_i)$ and $y \leftarrow \min(y, b_i)$.
- Update $x \leftarrow \max(x, a_i)$ and $y \leftarrow \max(y, b_i)$.

After performing all $(n-1)$ operations, Little Cyan Fish ends up with a final tuple (x, y). The challenge is to determine, for each value of k from 0 to $(n-1)$, the count of all distinct final tuples (x, y) such that $|x - y| = k$.

Input

There are multiple test cases in a single test file. The first line of the input contains a single integer T ($1 \le T \le 10^5$), indicating the number of test cases.

For each test case, the first line of the input contains a single integer n ($2 \le n \le 5 \times 10^5$).

The next line of the input contains n integers $a_1, a_2, \cdots, a_n$ ($1 \le a_i \le n$, $a_i \ne a_j$ for all $1 \le i < j \le n$).

The next line of the input contains n integers $b_1, b_2, \cdots, b_n$ ($1 \le b_i \le n$, $b_i \ne b_j$ for all $1 \le i < j \le n$).

It is guaranteed that the sum of n over all test cases does not exceed 5×10^5.

Output

For each test case, output a single line contains n integers. The i-th integer represents the answer for $k = i - 1$.

Example

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

Problem F. Whose Land?

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

Little Cyan Fish, also known as Xiao Qingyu, is the president of the Memoria Land. The Memoria Land consists of n cities, labeled by the integers from 1 to n. There are $n-1$ roads, where the i-th road connects cities u_i and v_i. It is guaranteed that for any two cities there is a path connecting them. In other words, the Memoria Land can be treated as a tree with n vertices. The distance between two cities x and y, denoted by $dis(x, y)$, is defined as the number of edges in the shortest path connecting these two cities. Specially, we have $dis(x, x) = 0$, as an empty path will meet the requirement.

As the anniversary of Memoria Land draws near, Little Cyan Fish is planning to show movies in selected cities to commemorate this special time. For the residents of city x, they can travel to city y to see a movie if and only if the city y is showing movies, and $dis(x, y) \le k$.

Now the Little Cyan Fish wishes to make plans to show the movie. He has a total of q questions, and for the i-th question, he wants to know how many cities will be able to see the movie, if the movies are shown in all cities numbered from l_i to r_i.

Well, ah, handling all the tasks on the Memoria Land had made the Little Cyan Fish so tired that he no longer had the strength to answer these questions. Please help the Little Cyan Fish by answering all these questions.

Input

There are multiple test cases in a single test file. The first line of the input contains a single integer T ($1 \le T \le 10^5$), indicating the number of test cases.

For each test case, the first line of the input contains three integers n, k, and q ($1 \le n \le 10^5$, $1 \le k \le 20$, $1 \le q \le 5 \times 10^5$).

For the following $(n-1)$ lines, the i-th line contains two integers u_i and v_i ($1 \le u_i, v_i \le n$), indicating an edge connecting vertices u_i and v_i in the tree.

The next q lines describes all the queries. The i-th line of these lines contains two integers l_i and r_i ($1 \le l_i \le r_i \le n$), indicating a query.

It is guaranteed that:

- The sum of n over all test cases does not exceed 5×10^5.
- The sum of q over all test cases does not exceed 5×10^5.
- The sum of $n \cdot k$ over all test cases does not exceed 2×10^6.

Output

For each query, output a single line contains a single integer, indicating the answer.

Example

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

Problem G. China Convex Polygon Contest

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

Kevin and Little Cyan Fish participated in the China Convex Polygon Competition Final (CCPC Final). The online judging system features a statistic called "Last Success", indicating who is currently the last person to successfully solve a problem.

The competition lasts for m seconds and comprises n problems. Little Cyan Fish learned that Kevin will solve a problem at seconds $a_1, a_2, \cdots, a_n$ after the contest starts. And it takes Little Cyan Fish $b_1, b_2, \cdots, b_n$ seconds to complete each problem. Little Cyan Fish wants to devise a strategy (i.e. the order of solving the problems and when to submit) that maximizes the duration for which he is the last to achieve success.

Please note that:

- After finishing a problem, Little Cyan Fish is not required to submit it immediately. He can choose to work on solving another problem in the meantime.
- Submissions and other operations do not consume time, allowing Little Cyan Fish to submit a problem while completing another.
- If Little Cyan Fish and Kevin submit solutions at the same time, the Last Success is attributed to Little Cyan Fish.
- The Last Success is empty when the contest begins and before any submissions.

Input

There are multiple test cases in a single test file. The first line of the input contains a single integer T ($1 \le T \le 10^5$), indicating the number of test cases.

For each test case, the first line of the input contains two integers n and m ($1 \le n \le 10^5, 1 \le m \le 10^9$).

The next line of the input contains n integers $a_1, a_2, \cdots, a_n$ ($1 \le a_i \le m$). It is guaranteed that $a_i < a_{i+1}$ for all $1 \le i < n$.

The next line of the input contains n integers $b_1, b_2, \cdots, b_n$ ($1 \le b_i \le m$).

It is guaranteed that the sum of n over all test cases does not exceed 10^5.

Output

For each test case, output a single line contains a single integer, indicating the answer.

Example

<table>
<tr><th>standard input</th><th>standard output</th></tr>
<tr><td><pre>3
3 10
1 5 9
1 2 3
3 10
1 5 9
1 1 4
3 10
1 5 9
1 5 10</pre></td><td><pre>9
9
7</pre></td></tr>
</table>

Note

For the first test case, Little Cyan Fish can:

- Starting from second 0, complete the first task at second 1 and submit it immediately;
- Starting from second 1, complete the second task at second 3 and wait until second 5 to submit it;
- Starting from second 4, complete the third task at second 7 and wait until second 9 to submit it;

Starting from second 1, the Last Success is Little Cyan Fish, so the answer is $10 - 1 = 9$.

For the third test case, Little Cyan Fish can:

- Starting from second 0, complete the first task at second 1 and submit it immediately;
- Starting from second 1, complete the second task at second 6 and submit it immediately;
- Give up the third task that can not be completed.

From second 1 to second 5, and from second 6 to second 9, the Last Success is Little Cyan Fish, so the answer is $(5 - 1) + (9 - 6) = 7$.

Problem H. The Game

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

After participating in the China Convex Polygon Competition Final (CCPC Final), Kevin and Little Cyan Fish decided to play a new game.

Kevin has an integer sequence $a_1, a_2, \ldots, a_{2n}$ of length $2n$. He and Little Cyan Fish will take turns to remove an element of the sequence, the remaining elements will be concatenated to form a new sequence. Kevin goes first. The game ends when there is only one element in the sequence. Kevin dislikes palindromes, therefore, if at any time during the process (including the initial sequence), the sequence is a palindrome, Little Cyan Fish wins. If the sequence has only one element left before being a palindrome, Kevin wins.

If Kevin and Little Cyan Fish both play optimally, who will be the winner?

A sequence $b_1, b_2, \ldots, b_m$ is palindrome if and only if for each $1 \le i \le m$, the condition $b_i = b_{m+1-i}$ is satisfied.

Input

There are multiple test cases in a single test file. The first line of the input contains a single integer T ($1 \le T \le 10^4$), the number of test cases.

For each test case, the first line of the input contains an integer n ($1 \le n \le 10^6$), and the second line contains $2n$ integers $a_1, a_2, \ldots, a_{2n}$ ($1 \le a_i \le 2n$), denoting the integer sequence.

It is guaranteed that the sum of n over all test cases does not exceed 10^6.

Output

For each test case, output a single line "`Kevin`" if the winner is Kevin, or "`Qingyu`" if the winner is Little Cyan Fish.

Example

standard input	standard output
3 3 1 1 4 5 1 4 2 1 2 3 4 4 1 2 2 3 2 1 1 4	Qingyu Kevin Qingyu

Problem I. All the Way Left

Input file: `standard input`
Output file: `standard output`
Time limit: 2.5 seconds
Memory limit: 1024 megabytes

Little Drink Congee is a good friend of Little Cyan Fish and a member of the famous acting group *All the Way Left*. He has recently been practicing his ability to recognize directions on a stage. To practice, he has selected n **distinct** points $A_1, A_2, \cdots, A_n$ on the stage. The stage is represented as a two-dimensional Cartesian plane, where the i-th point is located at the coordinates (x_i, y_i). Little Drink Congee aims to travel through all these points in the order of $p_1, p_2, \cdots, p_n$. A traversal is a permutation p of length n, where each point A_{p_i} is connected to $A_{p_{i+1}}$ with an oriented line segment.

Little Drink Congee thinks a traversal is considered good if and only if the following condition holds:

- It is non-self-intersecting, i.e. no two segments intersect except for two adjacent segments intersecting at one common endpoint.
- For each $1 \le i \le n-2$, the i-th turn is left (or going straight). Formally, the cross product of $\overrightarrow{A_{p_i}A_{p_{i+1}}}$ and $\overrightarrow{A_{p_{i+1}}A_{p_{i+2}}}$ is non-negative.

Little Drink Congee wants to know the number of good traversals, modulo $(10^9 + 7)$. However, he needs to spend time with Little Cyan Fish and cannot solve this challenge himself. Please help him calculate it!

Input

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

For each test case, the first line contains a single integer n ($1 \le n \le 2 \times 10^3$).

In the next n lines, the i-th line contains two integers x_i and y_i ($1 \le x_i, y_i \le 10^9$), denoting the coordinates of A_i. The coordinates of all the points are distinct.

It is guaranteed that the sum of n^2 over all test cases does not exceed 4×10^6.

Output

For each test case, output one line, containing the number of good traversals modulo $(10^9 + 7)$.

Example

standard input	standard output
3 4 1 1 3 1 2 2 2 3 3 1 1 1 2 1 3 6 1 1 2 1 2 2 2 3 3 2 4 2	6 2 13

Problem J. DFS Order 5

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

Stop, Yesterday Please No More.

Little Cyan Fish has a tree with n vertices. Each vertex is labeled from 1 to n. Now he wants to start a depth-first search at the vertex 1. The DFS order is the order of nodes visited during the depth-first search. A vertex appears in the j-th $(1 \le j \le n)$ position in this order means it is visited after $j-1$ other vertex. Because sons of a node can be iterated in arbitrary order, multiple possible depth-first orders exist.

The following pseudocode describes the way to generate a DFS order. The function GENERATE(x) returns a DFS order starting at vertex x:

Algorithm 1 An implementation of depth-first search

```
procedure DFS(vertex x)
    Append x to the end of dfs_order
    for each son y of x do                    ▷ Sons can be iterated in arbitrary order.
        DFS(y)                                ▷ The order might be different in each iteration.
    end for
end procedure
procedure GENERATE(x)
    Root the tree at vertex x
    Let dfs_order be a global variable
    dfs_order ← empty list
    DFS(x)
    return dfs_order
end procedure
```

Little Cyan Fish conducted Q depth-first searches on the entire tree, obtaining a DFS order each time. Unfortunately, Little Cyan Fish has a limited memory, and he only remembers a segment of each DFS order. Even more unfortunately, Little Cyan Fish cannot be sure his memory is correct. For each segment, he only remembers k numbers $a_1, a_2, \ldots, a_k$. He wants to ask for your help: is there a DFS order that satisfies $a_1, a_2, \ldots, a_k$ being a contiguous subsegment of this DFS order?

Input

The first line of the input contains two integers n and Q $(1 \le n, Q \le 10^5)$.

For the following $(n-1)$ lines, the i-th line contains two integers u_i and v_i $(1 \le u_i, v_i \le n)$, indicating an edge connecting vertices u_i and v_i in the tree.

The next q lines describes all the queries. The i-th line of these lines will first contain an integer k_i $(k_i \ge 1)$, and then k_i integers $a_1, a_2, \cdots, a_{k_i}$ $(1 \le a_i \le n)$, indicating a query.

It is guaranteed that the sum of k_i over all queries does not exceed 10^6.

Output

For each query, output a single line "`Yes`" or "`No`", indicating the answer.

Example

standard input	standard output
6 7 1 2 1 3 2 4 3 5 2 6 2 4 1 2 4 2 2 4 3 2 4 4 2 4 5 2 4 6 6 1 2 6 4 3 5	No No Yes No No Yes Yes

Problem K. Sticks

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

Little Cyan Fish has an $n \times n$ matrix. Each row and each column has a stick on its left side and top side, respectively. Let x_i represent the length of the stick on the left side of the i-th row and y_i represent the length of the stick on the top side of the i-th column, where $0 \le x_i, y_i \le n$ and both are integers. Additionally, the sticks must not intersect, meaning there should be **no** $i, j \in [1, n]$ such that both $x_i \ge j$ and $y_j \ge i$ hold true.

Little Cyan Fish defines the matrix A as follows:

- For each $i, j \in [1, n]$, if $x_i \ge j$ or $y_j \ge i$, then $A_{i,j} = 1$; otherwise, $A_{i,j} = 0$.

Given an $n \times n$ matrix M containing `0`s, `1`s, and `?`s, you need to determine how many different matrices can be formed by replacing each `?` with either a `0` or a `1`, so that there is at least one set of the sticks $\{x_1, x_2, \cdots, x_n, y_1, y_2, \cdots, y_n\}$ that could obtain this matrix. Since the answer may be large, output it modulo 998 244 353.

Input

The first line of the input contains a single integer n ($1 \le n \le 3\,000$).

The next n lines of the input describes the matrix M. Each of the line contains a string of length n containing "`0`", "`1`", and "`?`", indicating the matrix.

Output

Output a single line contains a single integer, indicating the answer.

Examples

standard input	standard output
2 ?? ??	14
5 ??1?? ?1??0 ??0?? ???1? ??1??	3144
10 0000000000 ?????????? ?????????? ?????????? ?????????? ?????????? ?????????? ?????????? ?????????? ??????????	361458985

Problem L. Exchanging Kubic

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

This is an interactive problem.

In computer science, the maximum sum subarray problem, also known as the maximum segment sum problem, is the task of finding a contiguous subarray with the largest sum, within a given one-dimensional array $A_1, A_2, \cdots, A_n$ of numbers. Formally, the task is to find indices i and j, such that the following sum is as large as possible:

$$\sum_{i \le k \le j} A_k$$

It is also possible to choose an empty subarray, which means you found an empty array with the sum 0. The value of the maximum sum subarray is denoted by $MSS(A)$. For example, $MSS([-2, 1, 4, -3, 5]) = 7$, $MSS([-5]) = 0$, and $MSS([-1, -2]) = 0$.

Little Cyan Fish is taking a course on Kubic Theory at Powerful Kubic University (PKU). Today, Prof. Kubic asked Little Cyan Fish to play the following game with him during the course:

Prof. Kubic has a sequence of integers $a_1, a_2, \cdots, a_n$. Little Cyan Fish could ask Prof. Kubic at most $2n$ questions in the following form:

- `? l r`: query the value of $MSS(a_l, a_{l+1}, \cdots, a_r)$.

The task of Little Cyan Fish is to report a sequence $b_1, b_2, \cdots, b_n$ satisfying:

- $MSS(a_l, a_{l+1}, \cdots, a_r) = MSS(b_l, b_{l+1}, \cdots, b_r)$ for all $1 \le l \le r \le n$.

Little Cyan Fish found this task is very challenging. Can you help him to develop a strategy to finish Prof. Kubic's task?

Interaction Protocol

There are multiple test cases in a single test file. The first line of the input contains a single integer T ($1 \le T \le 10^4$), indicating the number of the test cases.

For each test case, the first line of the input contains a single integer n ($1 \le n \le 2\,000$).

Then, the interaction begins. You may perform at most $2n$ queries in each test case. To perform a query, you need to print a single line "`?` l r" ($1 \le l \le r \le n$), indicating a query. Then, you need to read the result of your query from the standard input.

To give your answer, you need to print "`!` b_1 b_2 $\cdots$ b_n". You need to ensure that $-10^{15} \le b_i \le 10^{15}$. Printing the answer is not considered a query and does not count toward the $2n$ limit. After printing you answer, you need to read the next case, or terminate your program immediately.

After printing a query, do **NOT** forget to output end of line and flush the output. To do this, use "`fflush(stdout)`" or "`cout.flush()`" in C++, "`System.out.flush()`" in Java, "`flush(output)`" in Pascal, or "`stdout.flush()`" in Python.

It is guaranteed that $-10^9 \le a_i \le 10^9$, and the sum of n over all test cases does not eceed 10^4.

In this problem, it is guaranteed that the interactor is **non-adaptive**. That is, the values of a_i are decided before the interaction process. They will not be changed based on your queries.

Example

standard input	standard output
2	
3	
	? 1 1
1	
	? 2 2
0	
	? 3 3
1	
	! 1 -1 1
5	
	? 1 3
4	
	? 3 5
5	
	! 2 -1 3 -4 5

Problem M. Bot Friends

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

Little Cyan Fish has n robots friends (referred to as bots) positioned on a number line. The i-th bot is located at position $i + 0.5$ for $1 \le i \le n$. Additionally, there are $n + 1$ holes on the number line, with the i-th hole at position i for $1 \le i \le n + 1$.

Initially, all bots are inactive. The Little Cyan Fish activates the bots sequentially in any order. Upon activation, a bot moves either to its left or right, following these rules based on its type:

- Bots of type “<” can only move left.
- Bots of type “>” can only move right.
- Bots of type “?” can choose to move either left or right.

A bot continues to move in its chosen direction until it falls into an unoccupied hole, at which point it stops and occupies that hole. If a bot reaches the position 0 or the position $n + 2$, it disappears.

The objective is to avoid having bad bots, defined as follows: If a bot at position $i + 0.5$ falls into the hole at position i or $i + 1$, it is considered bad. Otherwise, it is considered good.

The challenge for the Little Cyan Fish is to determine the direction for each bot of type `?` and the activating order such that the number of good bots is maximized. Help Little Cyan Fish to find an optimal to assign the directions and activate the bots, and find the maximum number of good bots.

Input

There are multiple test cases in a single test file. The first line of the input contains a single integer T ($1 \le T \le 10^5$), indicating the number of test cases.

For each test case, the first line of the input contains a string of length n ($1 \le n \le 5\,000$). Each character of the string is either “<”, “>” or “?”, indicating the type of the i-th bot.

It is guaranteed that the sum of n^2 over all test cases does not exceed 5×10^7.

Output

For each test case, output a single line contains a single integer, indicating the answer.

Example

standard input	standard output
10	2
?>?	2
>?<	3
??<?	4
?><?<	5
??????	8
>?<?<>?<?<	7
?><???><><	8
??>>><><??	5
<>>?>>?>?>	6
<?<>>??<?>	