The 3rd Universal Cup

Stage 20: Kunming

December 7-8, 2024

This problem set should contain 13 problems on 26 numbered pages.

Based on

International Collegiate Programming Contest (ICPC)

Hosted by

Problem A. Antivirus

Time limit: 2 seconds
Memory limit: 1024 megabytes

In the introduction to network flow, the teacher had barely begun explaining when their pet chicken figured out how to solve the problem. The teacher exclaimed: Our avian flu is amazing!

—Roasted-chicken

Hyrule is entering the flu season.

Hyrule consists of n cities, which are connected by m **directed** roads. The capital (city numbered 1) can be reached from all cities through these roads.

The flu season will last for q days. On the i-th day at noon, the flu virus will start spreading from city a_i. The virus will spread along the roads to other cities. The virus spreads very quickly, so it can be assumed that all cities reachable from a_i will be infected immediately. As the capital city of Hyrule, city 1 is crucial for the whole country. If the virus that starts spreading on the i-th day reaches the capital, there will be an economic loss cost of b_i.

To prevent the virus from spreading, Auru can deploy a virus filter in a city each night (including the night before the first day). Deploying a virus filter in city i has a deployment cost of c_i. The virus cannot spread through the city equipped with a virus filter, and that city will also remain uninfected. If the virus were to start spreading from a city equipped with a virus filter, the virus would simply disappear without infecting any cities. The deployed virus filter remains effective until Auru deploys a new filter in another city. (In other words, there can be at most one virus filter at a time.)

Auru wants to know: for each $i = 1, 2, \ldots, q$, considering only the viruses in the first i days, what is the minimum value of "economic loss cost + deployment cost".

Input

Each test file contains multiple test cases. The first line contains the number of test cases T ($1 \le T \le 10^4$). The description of the test cases follows.

The first line contains three integers n, m, and q ($2 \le n \le 10^5$, $n - 1 \le m \le 2 \times 10^5$, $1 \le q \le 10^5$), representing the number of cities, the number of roads, and the duration of the flu season in days, respectively.

In the next m lines, each contains two integers u_i and v_i ($1 \le u_i, v_i \le n$), representing a **directed** road from city u_i to city v_i. It is guaranteed that there are no self-loops, but **there may be multiple edges** between cities. The capital can be reached from all cities.

The next line contains n integers $c_1, c_2, \ldots, c_n$ ($1 \le c_i \le 10^9$), representing the deployment cost of the virus filter in each city.

In the following q lines, the i-th one contains two integers a_i and b_i ($2 \le a_i \le n$, $1 \le b_i \le 10^9$), indicating the city where the virus starts to spread on the i-th day and the economic loss cost if the virus reaches the capital.

For each test file, it is guaranteed that the sum of n over all test cases does not exceed 10^5, the sum of m over all test cases does not exceed 2×10^5, and the sum of q over all test cases does not exceed 10^5.

Output

For each test case, output a line containing n integers, where the i-th integer represents the minimum value of the "economic loss cost + deployment cost" when only considering the viruses in the first i days.

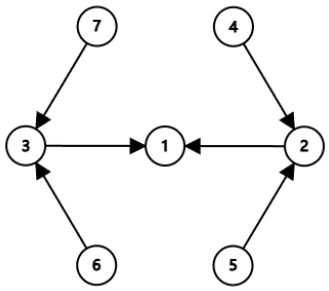
Example

standard input	standard output
3 7 6 4 2 1 3 1 4 2 5 2 6 3 7 3 4 3 5 2 2 1 1 4 2 5 2 6 2 7 2 5 6 4 1 3 3 2 2 1 4 2 5 4 2 5 10000 10000 2 100 5 5 1000 4 1000 3 1000 4 1000 4 4 1 2 1 3 1 4 2 4 3 100 1 1 100 4 10	2 3 4 4 5 100 102 202 10

Note

In the first test case:

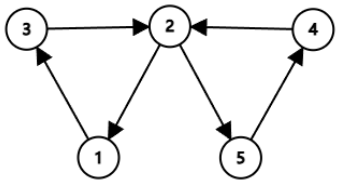
- The minimum cost after day 1 is 2: on night 0, deploy a virus filter in city 4 with a deployment cost of 2.
- The minimum cost after day 2 is 3: on night 0, deploy a virus filter in city 2 with a deployment cost of 3.
- The minimum cost after day 3 is 4: on night 0, deploy a virus filter in city 2 with a deployment cost of 3; on night 2, deploy a virus filter in city 6 with a deployment cost of 1.
- The minimum cost after day 4 is 4: on night 0, deploy a virus filter in city 1 with a deployment cost of 4.



Example 1 Illustration

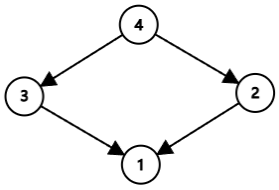
In the second test case:

- The minimum cost after day 1 is 5: on night 0, deploy a virus filter in city 5 with a deployment cost of 5.
- The minimum cost after day 2 is 100: on night 0, deploy a virus filter in city 4 with a deployment cost of 100.
- The minimum cost after day 3 is 102: on night 0, deploy a virus filter in city 4 with a deployment cost of 100; on night 2, deploy a virus filter in city 3 with a deployment cost of 2.
- The minimum cost after day 4 is 202: on night 0, deploy a virus filter in city 4 with a deployment cost of 100; on night 2, deploy a virus filter in city 3 with a deployment cost of 2; on night 3, deploy a virus filter in city 4 again with a deployment cost of 100.



Example 2 Illustration

In the third test case, since only one virus filter can exist at a time, it is not possible to stop the virus from spreading by deploying virus filters in both cities 2 and 3. The deployment costs for cities 1 and 4 are higher than the economic loss cost caused by the virus, so no virus filter is deployed, and the minimum cost is 10 (economic loss cost).



Example 3 Illustration

Problem B. Brackets

Time limit: 3 seconds
Memory limit: 1024 megabytes

Originally, there were only three kinds of brackets. But due to the harsh realities of life, the greater-than and less-than signs started moonlighting as angle brackets.

—Stir-fried-chicken

Betty has a bracket sequence s of length n, which consists of eight types of brackets: "`()[]{}<>`". Additionally, she has m sub-bracket sequences, and the i-th sub-bracket sequence t_i is the sequence formed by concatenating the characters from the l_i-th to the r_i-th position of s, i.e., $t_i = s_{l_i} s_{l_i+1} \cdots s_{r_i}$.

Betty wants to know, by taking out several pairs from these m sub-bracket sequences and concatenating sequences in the same pair, how many valid bracket sequences† can be formed at most? Formally, you need to find as many pairs (a_i, b_i) as possible such that:

1. For any integer x from 1 to m, the sum of its occurrences in a and its occurrences in b should be no larger than 1.
2. For any pair (a_i, b_i), $t_{a_i} t_{b_i}$ is a valid bracket sequence†.

† A bracket sequence x is valid if and only if it satisfies one of the following conditions:

- $|x| = 0$, i.e., x is an empty string.
- $x = LyR$, where y is a valid bracket sequence, and L, R are left and right brackets of the same type (i.e., $L =$ "`(`" and $R =$ "`)`", or $L =$ "`[`" and $R =$ "`]`", or $L =$ "`{`" and $R =$ "`}`", or $L =$ "`<`" and $R =$ "`>`").
- $x = y_1 y_2$, where both y_1 and y_2 are valid bracket sequences.

Input

Each test file contains multiple test cases. The first line contains the number of test cases T ($1 \le T \le 10^4$). The description of the test cases follows.

The first line contains two integers n and m ($1 \le n, m \le 5 \times 10^5$), which represent the length of s and the number of sub-bracket sequences, respectively.

The second line contains a string s ($|s| = n$), which consists only of the eight brackets "`()[]{}<>`".

The next m lines each contain two integers l_i and r_i ($1 \le l_i \le r_i \le n$).

For each test file, it is guaranteed that the sum of n over all test cases does not exceed 5×10^5, and the sum of m over all test cases does not exceed 5×10^5.

Output

For each test case, output a single integer on a new line, representing the maximum number of valid bracket sequences that can be obtained by concatenating pairs of sub-bracket sequences.

Example

standard input	standard output
4 8 1 ()[]{}<> 3 6 2 6)(1 1 1 1 1 1 2 2 2 2 2 2 6 2 ([)(]) 1 3 4 6 22 8 ([{}<<<<])>>>>([]){()} 3 8 11 14 1 10 3 4 19 22 20 21 17 20 21 22	0 3 0 2

Note

In the first test case, t_1 = "[]{}". Although t_1 is already a valid bracket sequence, there are no other sub-bracket sequences that can be paired with it, so the output is 0.

In the second test case, there are six sub-bracket sequences containing three "(" and three ")", which can be used to form three valid bracket sequences "()". Therefore, the output should be 3.

In the third test case, no valid bracket sequences can be formed.

In the fourth test case, one possible matching is t_1t_2: "{}<<" + ">>", and t_4t_5: "{}" + "{()}".

Problem C. Coin

Time limit: 2 seconds
Memory limit: 1024 megabytes

The government decided to cut the navy's budget and distribute the saved gold directly to the pirates. This managed to eliminate more pirates than the navy ever did.

—Boiled-chicken

The pirates have just seized a giant gold coin!

To determine the ownership of this gold coin, they decided to select the owner using the following method:

Let the current number of remaining pirates be n. The pirates line up in a queue, and the pirates at positions $1, 1+k, 1+2k, \ldots, 1+(\lceil \frac{n}{k} \rceil - 1)k$ are eliminated. This operation is repeated until only one pirate remains. The final remaining pirate will receive the gold coin.

Charlie is the smartest among the pirates. He wants to know where he should stand initially to be the last pirate remaining and win the coin.

Input

Each test file contains multiple test cases. The first line contains the number of test cases T ($1 \le T \le 100$). The description of the test cases follows.

The first line of each test case contains two integers n and k ($2 \le n, k \le 10^{12}$), representing the initial number of pirates and the parameter used for elimination.

For each test file, it is guaranteed that the sum of n over all test cases does not exceed 10^{12}, and the sum of k over all test cases does not exceed 10^{12}.

Output

For each test case, output a single integer on a new line, indicating the position of the pirate who will ultimately receive the gold coin in the initial queue.

Example

standard input	standard output
4 6 2 8 3 10000 2 1919810 114514	4 8 8192 1919805

Note

For the first test case in the example, the positions of the remaining pirates in the original sequence after each round are:

- Initial state: $1, 2, 3, 4, 5, 6$.
- After the first round: $2, 4, 6$.
- After the second round: 4.

For the second test case in the example, the positions of the remaining pirates in the original sequence after each round are:

- Initial state: $1, 2, 3, 4, 5, 6, 7, 8$.
- After the first round: $2, 3, 5, 6, 8$.
- After the second round: $3, 5, 8$.
- After the third round: $5, 8$.
- After the fourth round: 8.

Problem D. Dolls

Time limit: 1 second
Memory limit: 1024 megabytes

A little chicken was playing a ring-toss game and ended up tossing rings onto sets of nesting dolls.

—Steamed-chicken

David has just obtained n Russian nesting dolls of distinct sizes. He arranges these dolls in a row from left to right, where the i-th position contains a doll of size a_i.

Let the size of the smallest doll in the i-th position be l_i, and the size of the largest one be r_i. Dolls over two adjacent positions i and $i+1$ can be merged if and only if $r_i < l_{i+1}$ or $r_{i+1} < l_i$. The new nesting doll will contain all the dolls from the original i-th and $i+1$-th positions and will be placed in the i-th position. All dolls in positions greater than $i+1$ will shift left by one position to fill the gap.

For example, when $n = 4, a = [2, 1, 4, 3]$, David can:

1. Merge the dolls in positions 1 and 2. Now the remaining dolls have sizes $[(1, 2), (4), (3)]$.
2. Merge the dolls in positions 2 and 3. Now the remaining dolls have sizes $[(1, 2), (3, 4)]$.
3. Merge the dolls in positions 1 and 2. Now all the dolls have been merged into one position.

How many merge operations at most can David perform under an optimal strategy?

Input

Each test file contains multiple test cases. The first line contains the number of test cases T ($1 \le T \le 10^4$). The description of the test cases follows.

The first line of each test case contains an integer n ($1 \le n \le 10^5$), representing the number of nesting dolls.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le n$, $\forall i \ne j, a_i \ne a_j$), representing the initial sizes of the dolls in each position.

For each test file, 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 integer on a new line, representing the maximum number of merge operations that can be performed.

Example

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

Problem E. Extracting Weights

Time limit:	1 second
Memory limit:	1024 megabytes

"Emily is very busy"... I feel like Emily is actually pretty free.

—Grilled-chicken

This is an interactive problem.

Emily has a tree containing n nodes, where the weight of the node numbered i is w_i. Node 1 is the root node, and the weight of the root node is 0.

Emily wants you to guess the weight of each node. Specifically, you can make at most n queries, with the i-th query containing two integers u_i and v_i. If the simple path from u_i to v_i contains exactly k edges, Emily will tell you the bitwise XOR[†] of the weights of all the nodes on the simple path from node u_i to node v_i (including the endpoints). Otherwise, Emily will respond with "-1", indicating that she does not want to answer that question. Emily is very busy, so she will not start answering until you have asked all your questions.

Of course, you may not be able to guess the weights of all nodes with no more than n queries for some cases. In such cases, you should not make any queries and instead directly tell Emily that it is impossible.

[†]The bitwise XOR operation refers to the addition of each bit of two binary numbers under modulo 2. For example: $(0011)_2 \oplus (0101)_2 = (0110)_2$.

Input

There is only one test case in each test file.

The first line contains two integers n and k ($2 \le n \le 250$, $1 \le k \le n-1$), representing the number of nodes in the tree and a parameter mentioned in the query.

The next $n-1$ lines each contain two integers x_i and y_i ($1 \le x_i, y_i \le n$), indicating that there is an edge between nodes x_i and y_i.

It is guaranteed that all edges form a tree, and the correct answer satisfies $0 \le w_i < 2^{30}$.

Interaction Protocol

First, you need to determine whether it is possible to guess the weights of all nodes within n queries. If not, your program should output "`No`" and exit immediately. Otherwise, your program should output "`Yes`" and proceed to querying. 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.

To make a query, output **all** your q queries **at once** in the format "`?` q u_1 v_1 u_2 v_2 ... u_q v_q" ($1 \le q \le n$, $1 \le u_i, v_i \le n$). After flushing your output, your program should read a line containing q integers. The i-th integer represents the response to the i-th query.

To guess the weights, output your guess in the format "`!` w_2 w_3 ... w_n" ($0 \le w_i < 2^{30}$). After flushing your output, your program should exit immediately.

Note that the answer for each test case is pre-determined. That is, the interactor is not adaptive. Also, note that your guess does not count as a query.

To flush your output, you can use:

- `fflush(stdout)` (if you use `printf`) or `cout.flush()` (if you use `cout`) in C and C++.
- `System.out.flush()` in Java and Kotlin.

- `stdout.flush()` in Python.

Examples

standard input	standard output
4 1 1 2 2 3 2 4 1 3 2	 YES ? 3 1 2 2 3 2 4 ! 1 2 3
5 2 1 2 2 3 3 4 3 5 1 2 3 4	 YES ? 4 1 3 2 4 2 5 4 5 ! 4 5 3 2
6 2 1 2 2 3 3 4 4 5 4 6	 NO

Problem F. Flowers

Time limit: 3 seconds
Memory limit: 1024 megabytes

"Among Little Grass, Little Flower, Little Chicken, and Little Rabbit, who is better at network flow?"
"Little Flower, because it's a flower."

—Smoked-chicken

Frank is fascinated by the beauty of numbers.

One day, Frank was watering flowers in his garden. Looking at the beautiful petals, he thought it would be wonderful if each flower could grow numbers.

So, he took out paper and pen and started sketching his ideal "number flower". An undirected connected graph is called a "number flower" if and only if it satisfies the following three conditions:

1. If the graph contains n nodes, these nodes should be labeled from 1 to n.
2. The graph contains exactly $n-1$ edges. No node has a degree greater than 2 except for node 1.
3. Nodes directly connected to node 1 are called *key nodes*. All *key nodes* have pairwise coprime labels. For each *non-key node* (except node 1), its label is a multiple of the nearest *key node*'s label, and the labels along the simple path from the *non-key node* to its nearest *key node* are monotonically decreasing.

Given an integer n, how many different "number flowers" with n nodes are there? Two graphs G_1 and G_2 are considered the same if and only if for any edge (u, v) in G_1, a corresponding edge (u, v) exists in G_2.

Since the answer is huge, you only need to output it modulo a prime number p.

Input

There is only one test case in each test file.

The first line contains two positive integers n and p ($1 \le n \le 10^{10}$, $10^8 < p < 10^9$).

It is guaranteed that p is a prime number.

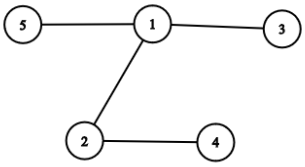
Output

Output a single integer, which is the number of different "number flowers" modulo p.

Examples

standard input	standard output
5 998244353	1
10 998244353	4
10000000000 998244353	889033323

Note



Example 1 Illustration

For the first example, there is only one "number flower", which is shown in the figure.

Problem G. GCD

Time limit: 3 seconds
Memory limit: 1024 megabytes

Does anyone understand why many problem setters think it is funny to place a GCD-themed problem as problem G?

—Braised-chicken

Given two integers a and b, you can perform one of the following two operations in each round:

- If $a > 0$, then reduce the value of a by $\gcd(a, b)$.
- If $b > 0$, then reduce the value of b by $\gcd(a, b)$.

Grace wants to know the minimum number of rounds needed to make both a and b become 0.

† $\gcd(x, y)$ denotes the greatest common divisor of x and y. For example, $\gcd(6, 8) = 2, \gcd(7, 5) = 1$. The values of $\gcd(x, 0)$ and $\gcd(0, x)$ are defined as x.

Input

Each test file contains multiple test cases. The first line contains the number of test cases T ($1 \le T \le 1000$). The description of the test cases follows.

Each test case consists of a single line containing two integers a and b ($1 \le a \le b, a \le 5000, b \le 10^{18}$).

For each test file, it is guaranteed that the sum of a over all test cases does not exceed 10^4.

Output

For each test case, output a single integer representing the minimum number of rounds needed to make both a and b become 0.

Example

standard input	standard output
3 3 4 12 20 114 514	3 4 6

Note

For the first test case in the example, one possible optimal solution is:

- Perform an operation on a: $a = 3 - \gcd(3, 4) = 2$.
- Perform an operation on a: $a = 2 - \gcd(2, 4) = 0$.
- Perform an operation on b: $b = 4 - \gcd(0, 4) = 0$.

Problem H. Horizon Scanning

Time limit: 1 second
Memory limit: 1024 megabytes

Cold joke: Radar is a palindrome.
Explanation: It's actually cold knowledge, not a cold joke. The confusion between these concepts is what makes it funny.

—Soy-chicken

Hana recently needs to develop a radar system to monitor abnormal activities across the archipelago she manages.

There are n islands in the ocean, and the i-th island is located at coordinates (x_i, y_i), which can be treated as a point on the plane. Assume the radar has a scanning angle range of α. When the radar is rotated to an angle θ, it can monitor all the islands located within the angular range $[\theta - \frac{\alpha}{2}, \theta + \frac{\alpha}{2}]$.

Hana is currently low on funds, so she wants to minimize the cost of building the radar. She wants to know, when the radar is placed at the origin $(0, 0)$, what the minimum scanning angle α should be to ensure that for any angle θ, the radar can monitor at least k islands.

Input

Each test file contains multiple test cases. The first line contains the number of test cases T ($1 \le T \le 10^4$). The description of each test case follows.

The first line of each test case contains two integers n and k ($1 \le k \le n \le 2 \times 10^5$), representing the total number of islands and the minimum number of islands the radar must monitor at any given time.

The next n lines each contain two integers x_i and y_i ($|x_i|, |y_i| \le 10^9$), representing the position of an island. It is guaranteed that the coordinates of any two islands are different and none of them are located at the origin.

For each test file, it is guaranteed that the sum of n over all test cases does not exceed 2×10^5.

Output

For each test case, output a single decimal fraction representing the minimum radar scanning angle in radians.

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 $\frac{|a-b|}{\max(1,|b|)} \le 10^{-6}$.

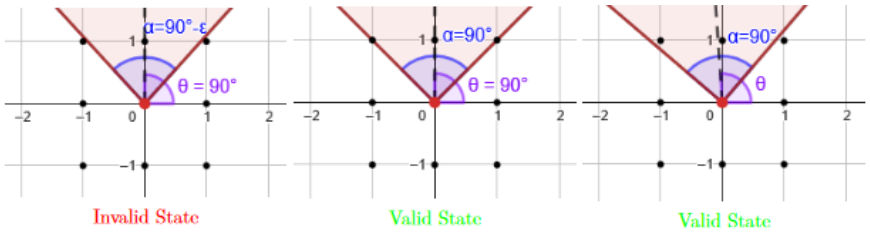
Example

standard input	standard output
5 1 1 0 1 8 2 1 0 1 1 0 1 -1 1 -1 0 -1 -1 0 -1 1 -1 4 2 -1 1 0 1 0 2 1 1 4 2 -1000000000 0 -998244353 1 998244353 1 1000000000 0 3 1 0 1 0 2 0 -1	6.2831853072 1.5707963268 5.4977871438 3.1415926546 3.1415926536

Note

For the first test case in the example, there is only one island on the plane at $(0, 1)$. To ensure that at least one island is always within range, the radar's monitoring range must be set to 360°, which is 2π in radians.

For the second test case in the example, there are 8 islands on the plane, with each pair of islands separated by 45°. If the radar's range is less than 90°, it would only be able to monitor one island when one of its boundaries just moves past an island (as shown on the left of the illustration). Therefore, the minimum radar range needed is 90°, ensuring that at least two islands are always within range.



Example 2 Illustration

Problem I. Items

Time limit: 8 seconds
Memory limit: 1024 megabytes

"Every new knapsack goes on and on..."

—Celine Chicken

Ivan has n types of items, with an infinite supply of each type. The weight of the i-th type of item is w_i.

Ivan wants to know whether it is possible to select exactly n items such that the total weight of the selected items equals m.

Input

Each test file contains multiple test cases. The first line contains the number of test cases T ($1 \le T \le 10^4$). The description of the test cases follows.

The first line contains two integers n and m ($1 \le n \le 10^5, 0 \le m \le n^2$).

The second line contains n integers $w_1, w_2, \cdots, w_n$ ($0 \le w_i \le n$).

For each test file, it is guaranteed that the sum of all n over all test cases does not exceed 10^5.

Output

For each test case, output "`Yes`" if the desired selection is possible; 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 5 25 0 0 0 0 5 5 11 4 4 4 5 5 5 0 1 2 3 4 5 5 25 0 1 2 3 4	Yes No No No

Note

In the first test case, you can select 5 items of the 5th type, with a total weight of 25, so you output "`Yes`".

In the second test case, it can be proven that there is no selection of items that results in a total weight of 11, so you output "`No`".

In the third test case, since you must select 5 items and the weight of the lightest item is 1, there is no way to select items with a total weight of 0, so you output "`No`".

In the fourth test case, since the heaviest item weighs 4, there is no way to select items with a total weight of 25, so you output "`No`".

Problem J. Just another Sorting Problem

Time limit: 1 second
Memory limit: 1024 megabytes

According to the problem statement, Bob is the one who lives longer.

—Hotpot-chicken

Jessica is a master of sorting algorithms, proficient in selection sort, insertion sort, bubble sort, and many others. Therefore, she decided to host a sorting competition.

The competition takes place on a permutation† p of length n, with two participants: Alice and Bob. The two players take turns performing operations, with the first player being decided by a coin toss. If the sequence is in ascending order after **any** player's turn, Alice wins immediately. If Alice cannot win within a finite number of turns, Bob is considered the winner.

On Alice's turn, she can choose any two positions i, j ($i \neq j, 1 \leq i, j \leq n$) in the permutation and swap p_i and p_j. On Bob's turn, he can select two adjacent positions $i, i+1$ ($1 \leq i < n$) and swap p_i and p_{i+1}. Neither player is allowed to skip their turn.

Given the permutation p and the name of the player who operates first, determine who will win the game if both players play optimally.

† A permutation of length n is an array consisting of n distinct integers from 1 to n in arbitrary order. For example, $[2, 3, 1, 5, 4]$ is a permutation, but $[1, 2, 2]$ is not a permutation (2 appears twice in the array), and $[1, 3, 4]$ is also not a permutation ($n = 3$ but there is a 4 in the array).

Input

Each test file contains multiple test cases. The first line contains the number of test cases T ($1 \leq T \leq 10^4$). The description of the test cases follows.

The first line of each test case contains an integer n ($2 \leq n \leq 10^5$) and a string s ($s \in \{\texttt{Alice}, \texttt{Bob}\}$), representing the length of the permutation and the name of the player who operates first.

The second line contains n integers $p_1, p_2, \cdots, p_n$ ($1 \leq p_i \leq n$), representing the permutation p. It is guaranteed that there is at least one position i such that $p_i \neq i$.

For each test file, it is guaranteed that the sum of all n over all test cases does not exceed 10^5.

Output

For each test case, output one line containing the winner's name. If Alice wins, print "`Alice`"; otherwise, print "`Bob`".

Example

standard input	standard output
3 2 Alice 2 1 3 Bob 1 3 2 10 Bob 1 2 3 4 5 6 7 8 10 9	Alice Bob Bob

Problem K. Key Recovery

Time limit: 1 second
Memory limit: 1024 megabytes

Did you know? During the selection of ISIJ contestants, ISTJs and ISFJs don't get extra points.

—Mixed-chicken

This is an interactive problem.

Kevin is dissatisfied with his MBTI, so he asked Doraemon for a magical machine: the MBTI converter.

The MBTI converter can convert the MBTI attributes of 8 people at a time. It will generate a "random" new MBTI for all individuals based on their initial MBTIs and a 32-bit key k.

Kevin carefully studied this machine and found that the "random" MBTI it generates actually follows a certain pattern. The machine mainly includes three basic structures that shuffle the MBTI: "`xor`", "`perm`", and "`mix`". Each of these structures takes the MBTIs of 8 individuals as input and outputs 8 MBTIs as a result. In each structure, the input 8 MBTIs are denoted as $in_0, in_1, in_2, \cdots, in_7$ and the output 8 MBTIs are denoted as $out_0, out_1, out_2, \cdots, out_7$. For convenience, Kevin replaces each MBTI with a hexadecimal number according to the following table.

INFP	INFJ	INTP	INTJ	ISFP	ISFJ	ISTP	ISTJ
0	1	2	3	4	5	6	7
ENFP	ENFJ	ENTP	ENTJ	ESFP	ESFJ	ESTP	ESTJ
8	9	a	b	c	d	e	f

"`xor`" Structure:

The `xor` structure flips certain attributes of the MBTI based on the key k.

Let k_j be the j-th binary bit of the key k ($0 \le j < 32$). The functionality of the `xor` structure can be expressed as:

$$out_i = in_i \oplus (\overline{k_{4i+3}k_{4i+2}k_{4i+1}k_{4i}})_2$$

where $\oplus$ denotes the bitwise XOR[1].

"`perm`" Structure:

The `perm` structure replaces the MBTI, where each MBTI is replaced by another fixed MBTI.

The functionality of the `perm` structure can be expressed as:

$$out_i = p_{in_i}$$

where p is a fixed array of length 16 with the following values:

p_0	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}	p_{11}	p_{12}	p_{13}	p_{14}	p_{15}
7	0	5	6	b	f	4	3	1	d	c	e	8	9	a	2

"`mix`" Structure:

The `mix` structure is used to mix the MBTIs of different people, with each output determined by two inputs.

The `mix` structure requires an intermediate array t of length 8, defined as:

$$t_i = \begin{cases} in_i \ll 1 & 0 \le in_i < 8 \\ ((in_i \oplus 8) \ll 1) \oplus 3 & 8 \le in_i < 16 \end{cases}$$

where $\oplus$ denotes the bitwise XOR[1] and $\ll$ denotes logical left shift of four binary digits[2].

The functionality of the `mix` structure can be expressed as:

$$out_i = in_{i \ggg 1} \oplus (t_{(i \ggg 1) \oplus 1})$$

where $\oplus$ denotes the bitwise XOR[1], and $\ggg$ denotes circular right shift of three binary digits[3].

The initial 8 MBTIs will sequentially go through 19 structures to produce the final output of 8 MBTIs. These structures are: `xor, perm, mix, xor, perm, mix, xor, perm, mix, xor, perm, mix, xor, perm, mix, xor, perm, mix, xor`, which is six consecutive rounds of "`xor, perm, mix`" followed by an "`xor`".

Users of this machine typically hope to obtain a specific MBTI. Unfortunately, when Doraemon built this machine, he encapsulated the key k inside the machine and refused to tell Kevin the value of the key. Thus, Kevin turned to you, the clever one. Kevin hopes you can construct some inputs (not exceeding 4096, or the machine may be damaged), test run to obtain the corresponding outputs, and then analyze the value of the key. This way, he can write a program to predict the output for each input, allowing everyone to get their desired MBTI.

Can you accomplish this task?

[1] Bitwise XOR refers to performing the modulo 2 addition operation on each corresponding bit of two binary numbers. For example: $(0011)_2 \oplus (0101)_2 = (0110)_2$.

[2] $x \ll y$ means to logically left shift the binary number x by y bits, discarding the overflow and filling the vacated parts with 0. For example: $(1100)_2 \ll 1 = (1000)_2$, $(1001)_2 \ll 1 = (0010)_2$.

[3] $x \ggg y$ means to circularly right shift the binary number x by y bits. For example: $(101)_2 \ggg 1 = (110)_2$, $(110)_2 \ggg 1 = (011)_2$.

Interaction Protocol

To run the MBTI converter, please output "`? ` $in_7in_6in_5in_4in_3in_2in_1in_0$" ($in_i \in \{0-9, a-f\}$), where in_i is the hexadecimal value corresponding to the i-th input MBTI. You may perform this operation at most 4096 times. After flushing your output, please read the output of the MBTI converter in the format "$out_7out_6out_5out_4out_3out_2out_1out_0$" ($out_i \in \{0-9, a-f\}$).

To answer for the value of k, please output "`! ` k" ($0 \le k < 2^{32}$), where k is the **decimal** representation of the value you believe the key k to be. You may perform this operation at most once. After performing the operation, you should exit your program immediately.

If you run the MBTI converter more than 4096 times, or if your output contains invalid characters, the interactor will output "`-1`". Upon receiving such a response, you should immediately exit your program, and you will receive a "`Wrong Answer`" result. Otherwise, the result is uncertain, as you will attempt to read data from a closed output stream.

To flush your output, you can use:

- `fflush(stdout)` (if you use `printf`) or `cout.flush()` (if you use `cout`) in C and C++.
- `System.out.flush()` in Java and Kotlin.
- `stdout.flush()` in Python.

Example

standard input	standard output
acf7e10b 105092e0	? 04f37255 ? 9090cfca ! 998244353

Note

The following information might be helpful when solving the problem:

1. There are no spaces between the in_is, out_is, so please do not output extra spaces. The output is **not** a hexadecimal number, so **do not** ignore leading 0s that may exist.
2. The range of k is $[0, 2^{32} - 1]$, so please be careful to avoid outputting negative numbers when outputting the answer.
3. The attachment for this problem includes "interact.hpp", which you may need for local debugging. For its usage, you can refer to the file "local_test_example.cpp" provided in the attachment.
4. The interactor is **not adaptive**. That is, the value of k in the interactor is fixed before the interaction begins and will not change according to your interactions.
5. There are 64 tests for this problem in total.

Problem L. Last Chance: Threads of Despair

Time limit: 2 seconds
Memory limit: 1024 megabytes

Dear problem setter, I downloaded Hearthstone. Why is my Threads of Despair a three-cost card?

—Pan-fried-chicken

Fried-chicken is a devoted player of Hearthstone. Since the game resumed operations in the Chinese mainland, he has been obsessed with it and reached Silver 2 rank in Standard mode. Today, while ranking up using Death Knight, he encountered a formidable opponent, Stewed-chicken, and was left with just 1 Health. To survive, Fried-chicken must eliminate all of Stewed-chicken's minions. Fortunately, he can use spell cards and his minions' attacks to achieve this goal.

Specifically, this game involves two factions: Fried-chicken and Stewed-chicken. Each faction has some minions. The i-th minion has h_i Health. It is now Fried-chicken's turn, and each of his minions can attack any one minion from the **opposing faction** at most once. When one minion attacks another, both minions lose 1 Health. If a minion's Health is reduced to 0 or less, it dies and can no longer attack or be attacked.

To achieve his goal, Fried-chicken casts the spell "Threads of Despair," causing every minion to explode upon death, which reduces the Health of **all** minions by 1. If the explosion causes the death of other minions, other minions will also explode immediately. Fried-chicken cannot have his minions attack other minions until all explosion effects have finished. After casting the spell, Fried-chicken can make his minions attack Stewed-chicken's minions in any order he chooses. He wants to know if there exists an attack order that allows Fried-chicken to eliminate all of Stewed-chicken's minions.

Input

Each test file contains multiple test cases. The first line contains the number of test cases T ($1 \le T \le 5 \times 10^5$). The description of the test cases follows.

The first line of each test case contains two integers n and m ($1 \le n, m \le 5 \times 10^5$), representing the number of Fried-chicken's minions and Stewed-chicken's minions, respectively.

The second line contains n integers $h_1, h_2, \ldots, h_n$ ($1 \le h_i \le 10^9$), where h_i represents the Health of Fried-chicken's i-th minion.

The third line contains m integers $h'_1, h'_2, \ldots, h'_m$ ($1 \le h'_i \le 10^9$), where h'_i represents the Health of Stewed-chicken's i-th minion.

For each test file, it is guaranteed that the sum of all n across all test cases does not exceed 5×10^5, and the sum of all m across all test cases does not exceed 5×10^5.

Output

For each test case, output "`Yes`" if Fried-chicken can eliminate all of Stewed-chicken's minions; 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.

Examples

standard input	standard output
3 3 2 1 1 4 2 6 3 2 1 1 4 2 7 2 1 100 100 2	Yes No Yes
3 7 1 1 1 1 1 1 1 1 9 5 2 3 4 5 6 7 1 6 5 3 3 4 5 6 7 1 5 7	No No Yes

Note

In the first test case of Sample 1, one possible sequence of actions is as follows: Fried-chicken's 3rd minion attacks Stewed-chicken's 2nd minion, followed by Fried-chicken's 2nd minion attacking Stewed-chicken's 2nd minion. At this point, Fried-chicken's 2nd minion dies, triggering an explosion. This explosion causes further deaths, leading to a chain reaction of explosions. Eventually, all minions are eliminated.

In the third test case of Sample 1, one possible sequence of actions is as follows: Fried-chicken's 1st minion attacks Stewed-chicken's 1st minion, followed by Fried-chicken's 2nd minion attacking Stewed-chicken's 1st minion. At this point, Stewed-chicken's 1st minion dies, triggering an explosion. Ultimately, Fried-chicken is left with two minions, each having 98 Health, while all of Stewed-chicken's minions are eliminated. Fried-chicken survives successfully.

Problem M. Matrix Construction

Time limit: 1 second
Memory limit: 1024 megabytes

Constructing a matrix, and the sample includes a test case "2 3"... Haven't we seen a problem like this recently?

—Stewed-chicken

Mary loves constructing matrices!

Today, Mary wants to fill a permutation† of length $n \times m$ into an $n \times m$ matrix A, such that the sum of any two adjacent elements is unique.

In other words, for any $1 \le x_1, x_2, x_3, x_4 \le n, 1 \le y_1, y_2, y_3, y_4 \le m$, if all of the following conditions are satisfied:

- $x_2 \ge x_1, y_2 \ge y_1, x_4 \ge x_3, y_4 \ge y_3$;
- $|x_2 - x_1| + |y_2 - y_1| = 1, |x_4 - x_3| + |y_4 - y_3| = 1$;
- $(x_1, y_1) \ne (x_3, y_3)$ or $(x_2, y_2) \ne (x_4, y_4)$;

then:

$$A_{x_1,y_1} + A_{x_2,y_2} \ne A_{x_3,y_3} + A_{x_4,y_4}$$

For example, when $n = 2$ and $m = 3$, matrix B is a valid solution, while matrix C is not valid because $C_{1,1} + C_{2,1} = C_{1,2} + C_{1,3}$.

$$B = \begin{bmatrix} 1 & 3 & 2 \\ 6 & 5 & 4 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

Given n and m, can all the conditions above be satisfied? If so, output a valid solution.

† A permutation of length n is an array consisting of n distinct integers from 1 to n in arbitrary order. For example, $[2, 3, 1, 5, 4]$ is a permutation, but $[1, 2, 2]$ is not a permutation (2 appears twice in the array), and $[1, 3, 4]$ is also not a permutation ($n = 3$ but there is a 4 in the array).

Input

Each test file contains multiple test cases. The first line contains the number of test cases T ($1 \le T \le 10^4$). The description of the test cases follows.

The first line contains two integers n, m ($1 \le n, m \le 1000$).

For each test file, it is guaranteed that the sum of $n \times m$ over all test cases does not exceed 10^6.

Output

For each test case, output on the first line whether a valid solution exists. If there is a valid solution, 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.

If a valid solution exists, you must also output n lines, each containing m integers. The j-th number on the i-th line represents the number at row i and column j in the matrix. You must make sure that the output numbers form a permutation of length $n \times m$. If there are multiple solutions, you may print any of them.

Example

standard input	standard output
2 1 1 2 3	yEs 1 YES 1 3 2 6 5 4