Codeforces Round #746 (Div. 2)

A. Gamer Hemose

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

One day, Ahmed_Hossam went to Hemose and said "Let's solve a gym contest!". Hemose didn't want to do that, as he was playing Valorant, so he came up with a problem and told it to Ahmed to distract him. Sadly, Ahmed can't solve it... Could you help him?

There is an Agent in Valorant, and he has n weapons. The i-th weapon has a damage value a_i, and the Agent will face an enemy whose health value is H.

The Agent will perform one or more moves until the enemy dies.

In one move, he will choose a weapon and decrease the enemy's health by its damage value. The enemy will die when his health will become less than or equal to 0. However, not everything is so easy: **the Agent can't choose the same weapon for 2 times in a row**.

What is the minimum number of times that the Agent will need to use the weapons to kill the enemy?

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 10^5$). Description of the test cases follows.

The first line of each test case contains two integers n and H ($2 \le n \le 10^3, 1 \le H \le 10^9$) — the number of available weapons and the initial health value of the enemy.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$) — the damage values of the weapons.

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

Output

For each test case, print a single integer — the minimum number of times that the Agent will have to use the weapons to kill the enemy.

Example

input
3 2 4 3 7 2 6 4 2 3 11 2 1 7
output
1 2 3

Note

In the first test case, the Agent can use the second weapon, making health value of the enemy equal to $4 - 7 = -3$. $-3 \le 0$, so the enemy is dead, and using weapon 1 time was enough.

In the second test case, the Agent can use the first weapon first, and then the second one. After this, the health of enemy will drop to $6 - 4 - 2 = 0$, meaning he would be killed after using weapons 2 times.

In the third test case, the Agent can use the weapons in order (third, first, third), decreasing the health value of enemy to $11 - 7 - 2 - 7 = -5$ after using the weapons 3 times. Note that we can't kill the enemy by using the third weapon twice, as even though $11 - 7 - 7 < 0$, it's not allowed to use the same weapon twice in a row.

B. Hemose Shopping

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Hemose was shopping with his friends Samez, AhmedZ, AshrafEzz, TheSawan and O_E in Germany. As you know, Hemose and his

friends are problem solvers, so they are very clever. Therefore, they will go to all discount markets in Germany.

Hemose has an array of n integers. He wants Samez to sort the array in the non-decreasing order. Since it would be a too easy problem for Samez, Hemose allows Samez to use only the following operation:

- Choose indices i and j such that $1 \leq i, j \leq n$, and $|i - j| \geq x$. Then, swap elements a_i and a_j.

Can you tell Samez if there's a way to sort the array in the non-decreasing order by using the operation written above some finite number of times (possibly 0)?

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \leq t \leq 10^5$). Description of the test cases follows.

The first line of each test case contains two integers n and x ($1 \leq x \leq n \leq 10^5$).

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($1 \leq a_i \leq 10^9$).

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

Output

For each test case, you should output a single string.

If Samez can sort the array in non-decreasing order using the operation written above, output "YES" (without quotes). Otherwise, output "NO" (without quotes).

You can print each letter of "YES" and "NO" in any case (upper or lower).

Example

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

Note

In the first test case, you can't do any operations.

In the second test case, the array is already sorted.

In the third test case, you can do the operations as follows:

- $[5, 1, 2, 3, 4]$, $swap(a_1, a_3)$
- $[2, 1, 5, 3, 4]$, $swap(a_2, a_5)$
- $[2, 4, 5, 3, 1]$, $swap(a_2, a_4)$
- $[2, 3, 5, 4, 1]$, $swap(a_1, a_5)$
- $[1, 3, 5, 4, 2]$, $swap(a_2, a_5)$
- $[1, 2, 5, 4, 3]$, $swap(a_3, a_5)$
- $[1, 2, 3, 4, 5]$

(Here $swap(a_i, a_j)$ refers to swapping elements at positions i, j).

C. Bakry and Partitioning

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Bakry faced a problem, but since he's lazy to solve it, he asks for your help.

You are given a tree of n nodes, the i-th node has value a_i assigned to it for each i from 1 to n. As a reminder, a tree on n nodes is a connected graph with $n - 1$ edges.

You want to delete **at least 1, but at most $k - 1$ edges** from the tree, so that the following condition would hold:

- For every connected component calculate the bitwise XOR of the values of the nodes in it. Then, these values have to be the same for all connected components.

Is it possible to achieve this condition?

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 5 \cdot 10^4$). Description of the test cases follows.

The first line of each test case contains two integers n and k ($2 \le k \le n \le 10^5$).

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$).

The i-th of the next $n - 1$ lines contains two integers u_i and v_i ($1 \le u_i, v_i \le n$, $u_i \neq v_i$), which means that there's an edge between nodes u_i and v_i.

It is guaranteed that the given graph is a tree.

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

Output

For each test case, you should output a single string. If you can delete the edges according to the conditions written above, output "YES" (without quotes). Otherwise, output "NO" (without quotes).

You can print each letter of "YES" and "NO" in any case (upper or lower).

Example

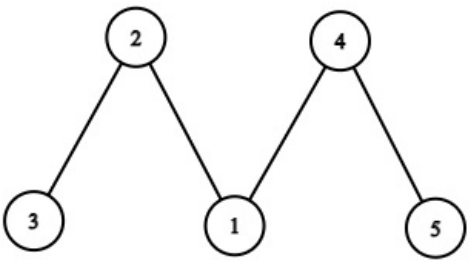
input
5 2 2 1 3 1 2 5 5 3 3 3 3 3 1 2 2 3 1 4 4 5 5 2 1 7 2 3 5 1 2 2 3 1 4 4 5 5 3 1 6 4 1 2 1 2 2 3 1 4 4 5 3 3 1 7 4 1 2 2 3
output
NO YES NO YES NO

Note

It can be shown that the objection is not achievable for first, third, and fifth test cases.

In the second test case, you can just remove all the edges. There will be 5 connected components, each containing only one node with value 3, so the bitwise XORs will be 3 for all of them.

In the fourth test case, this is the tree:



.

You can remove an edge $(4, 5)$

The bitwise XOR of the first component will be, $a_1 \oplus a_2 \oplus a_3 \oplus a_4 = 1 \oplus 6 \oplus 4 \oplus 1 = 2$ (where $\oplus$ denotes the bitwise XOR).

The bitwise XOR of the second component will be, $a_5 = 2$.

D. Hemose in ICPC ?

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

This is an interactive problem!

In the last regional contest Hemose, ZeyadKhattab and YahiaSherif — members of the team Carpe Diem — did not qualify to ICPC because of some ~~un~~known reasons. Hemose was very sad and had a bad day after the contest, but ZeyadKhattab is very wise and knows Hemose very well, and does not want to see him sad.

Zeyad knows that Hemose loves tree problems, so he gave him a tree problem with a very special device.

Hemose has a weighted tree with n nodes and $n - 1$ edges. Unfortunately, Hemose doesn't remember the weights of edges.

Let's define $Dist(u, v)$ for $u \neq v$ as the greatest common divisor of the weights of all edges on the path from node u to node v.

Hemose has a special device. Hemose can give the device a set of nodes, and the device will return the largest $Dist$ between any two nodes from the set. More formally, if Hemose gives the device a set S of nodes, the device will return the largest value of $Dist(u, v)$ over all pairs (u, v) with $u, v \in S$ and $u \neq v$.

Hemose can use this Device **at most** 12 **times**, and wants to find any two distinct nodes a, b, such that $Dist(a, b)$ is maximum possible. Can you help him?

Interaction

Begin the interaction from reading a single integer n ($2 \leq n \leq 10^3$) — the number of nodes in the tree.

Next, read $n - 1$ lines.

The i-th of the next $n - 1$ lines contains two integers u_i and v_i ($1 \leq u_i, v_i \leq n$, $u_i \neq v_i$), which means that there's an edge between nodes u_i and v_i.

It's guaranteed that weights of edges were $\leq 10^9$.

It is guaranteed that the given graph is a tree.

Now you may begin asking queries. To ask a query about a set of k nodes $v_1, v_2, \ldots, v_k$ ($2 \leq k \leq n$, $1 \leq v_i \leq n$, all v_i are distinct), output:

? k v_1 v_2 $\ldots$ v_k

You will then receive an integer x, the largest $Dist(v_i, v_j)$ over $1 \le i, j \le k$ with $i \neq j$.

When you have found a and b ($1 \le a, b \le n$), $a \neq b$) such that $Dist(a, b)$ is the maximum possible, print the answer in the following format:

! a b

Outputting answer doesn't count towards the limit of 12 queries.

If there are several pairs (a, b) with the same largest $Dist(a, b)$, you can output any.

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

- `fflush(stdout)` or `cout.flush()` in C++;
- `System.out.flush()` in Java;
- `flush(output)` in Pascal;
- `stdout.flush()` in Python;
- see the documentation for other languages.

Hacks

To hack a solution, use the following format.

The first line should contain a single integer n ($2 \le n \le 10^3$) — the number of nodes.

The i-th of the next $n - 1$ lines should contain three integers u_i, v_i, w_i ($1 \le u_i, v_i \le n$, $u_i \neq v_i$, $1 \le w \le 10^9$), which means that there's an edge between nodes u_i and v_i with weight w_i.

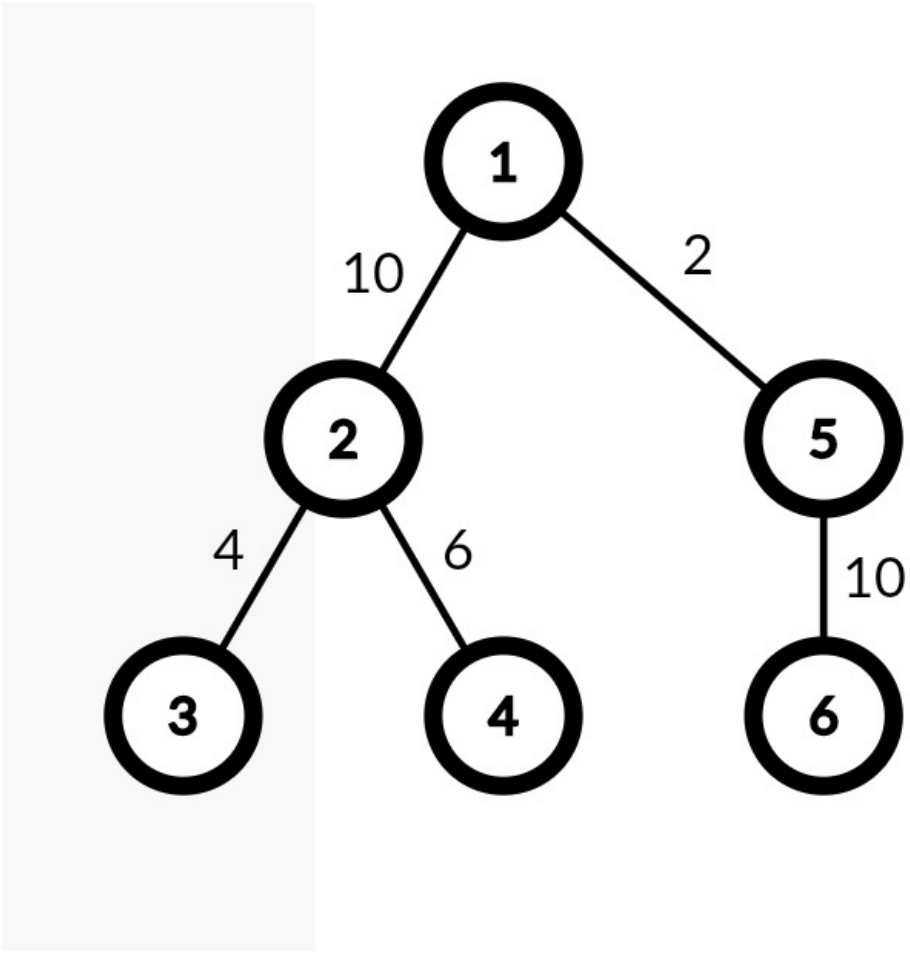
These $n - 1$ edges must form a tree.

Example

input
6 1 2 2 3 2 4 1 5 5 6 10 2 10
output
? 6 1 2 3 4 5 6 ? 3 3 1 5 ? 2 1 2 ! 1 2

Note

The tree in the first sample:



E. Bored Bakry

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Bakry got bored of solving problems related to xor, so he asked you to solve this problem for him.

You are given an array a of n integers $[a_1, a_2, \ldots, a_n]$.

Let's call a subarray $a_l, a_{l+1}, a_{l+2}, \ldots, a_r$ **good** if $a_l \,\&\, a_{l+1} \,\&\, a_{l+2} \,\ldots\, \&\, a_r > a_l \oplus a_{l+1} \oplus a_{l+2} \ldots \oplus a_r$, where $\oplus$ denotes the bitwise XOR operation and $\&$ denotes the bitwise AND operation.

Find the length of the longest good subarray of a, or determine that no such subarray exists.

Input

The first line contains a single integer n ($1 \le n \le 10^6$) — the length of the array.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^6$) — elements of the array.

Output

Print a single integer — the length of the longest good subarray. If there are no good subarrays, print 0.

Examples

input
2 5 6
output
2

input
3 2 4 3
output
0

input
6

8 1 3 3 1 2
output
4

Note
In the first case, the answer is 2, as the whole array is good: $5\&6 = 4 > 5 \oplus 6 = 3$.

In the third case, the answer is 4, and one of the longest good subarrays is $[a_2, a_3, a_4, a_5]$: $1\&3\&3\&1 = 1 > 1 \oplus 3 \oplus 3 \oplus 1 = 0$.

F1. Alice and Recoloring 1

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

The difference between the versions is in the costs of operations. Solution for one version won't work for another!

Alice has a grid of size $n \times m$, **initially all its cells are colored white**. The cell on the intersection of i-th row and j-th column is denoted as (i, j). Alice can do the following operations with this grid:

- Choose any subrectangle containing cell $(1, 1)$, and flip the colors of all its cells. (Flipping means changing its color from white to black or from black to white).

  This operation costs 1 coin.

- Choose any subrectangle containing cell $(n, 1)$, and flip the colors of all its cells.

  This operation costs 2 coins.

- Choose any subrectangle containing cell $(1, m)$, and flip the colors of all its cells.

  This operation costs 4 coins.

- Choose any subrectangle containing cell (n, m), and flip the colors of all its cells.

  This operation costs 3 coins.

As a reminder, subrectangle is a set of all cells (x, y) with $x_1 \le x \le x_2$, $y_1 \le y \le y_2$ for some $1 \le x_1 \le x_2 \le n$, $1 \le y_1 \le y_2 \le m$.

Alice wants to obtain her favorite coloring with these operations. What's the smallest number of coins that she would have to spend? It can be shown that it's always possible to transform the initial grid into any other.

Input
The first line of the input contains 2 integers n, m ($1 \le n, m \le 500$) — the dimensions of the grid.

The i-th of the next n lines contains a string s_i of length m, consisting of letters W and B. The j-th character of string s_i is W if the cell (i, j) is colored white in the favorite coloring of Alice, and B if it's colored black.

Output
Output the smallest number of coins Alice would have to spend to achieve her favorite coloring.

Examples

input
3 3 WWW WBB WBB
output
3

input
10 15 WWWBBBWBBBBBWWW BBBBWWWBBWWWBBB BBBWWBWBBBWWWBB BBWBWBBWWWBBWBW BBBBWWWBBBWWWBB BWBBWWBBBBBBWWW WBWWBBBBWWBBBWW WWBWWWWBBWWBWWW BWBWWBWWWWWWBWB BBBWBWBWBBBWWBW
output
74

Note
In the first sample, it's optimal to just apply the fourth operation once to the rectangle containing cells $(2,2), (2,3), (3,2), (3,3)$. This would cost 3 coins.

F2. Alice and Recoloring 2

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

The difference between the versions is in the costs of operations. Solution for one version won't work for another!

Alice has a grid of size $n \times m$, **initially all its cells are colored white**. The cell on the intersection of i-th row and j-th column is denoted as (i, j). Alice can do the following operations with this grid:

- Choose any subrectangle containing cell $(1, 1)$, and flip the colors of all its cells. (Flipping means changing its color from white to black or from black to white).

  This operation costs 1 coin.

- Choose any subrectangle containing cell $(n, 1)$, and flip the colors of all its cells.

  This operation costs 3 coins.

- Choose any subrectangle containing cell $(1, m)$, and flip the colors of all its cells.

  This operation costs 4 coins.

- Choose any subrectangle containing cell (n, m), and flip the colors of all its cells.

  This operation costs 2 coins.

As a reminder, subrectangle is a set of all cells (x, y) with $x_1 \le x \le x_2$, $y_1 \le y \le y_2$ for some $1 \le x_1 \le x_2 \le n$, $1 \le y_1 \le y_2 \le m$.

Alice wants to obtain her favorite coloring with these operations. What's the smallest number of coins that she would have to spend? It can be shown that it's always possible to transform the initial grid into any other.

Input
The first line of the input contains 2 integers n, m ($1 \le n, m \le 500$) — the dimensions of the grid.

The i-th of the next n lines contains a string s_i of length m, consisting of letters `W` and `B`. The j-th character of string s_i is `W` if the cell (i, j) is colored white in the favorite coloring of Alice, and `B` if it's colored black.

Output
Output the smallest number of coins Alice would have to spend to achieve her favorite coloring.

Examples

input
3 3 WWW WBB WBB
output
2

input
10 15 WWWBBBWBBBBBWWW BBBBWWWBBWWWBBB BBBWWBWBBBWWBB BBWBWBBWWWBBWBW BBBBWWWBBBWWBB BWBBWWBBBBBBWWW WBWWBBBBWWBBBWW WWBWWWWBBWWBWWW BWBWWBWWWWWWBWB BBBWBWBWBBBWWBW
output
68

Note
In the first sample, it's optimal to just apply the fourth operation once to the rectangle containing cells $(2,2), (2,3), (3,2), (3,3)$. This would cost 2 coins.