Winter Camp Contest 2022
Division 2

NYCU PCCA

ID	Problem Name	Time Limit
A	Apple Pen, Pineapple Pen	1 sec
B	Building a Kingdom	1 sec
C	Checkerboard Splitting	2 sec
D	DVD Player	1 sec
E	Elise Loves Drinks	1 sec
F	Frog on a Line	1 sec
G	Glass Key	5 sec
H	Hex Activation Key	6 sec
I	International Grandmaster	10 sec
J	Jujutsu Kaisen	2.5 sec
K	Kore wa FFT desu ka?	1 sec

Almost blank page

Problem A
Apple Pen, Pineapple Pen

Time limit: 1 second
Memory limit: 2048 megabytes

Problem Description

Einsplanck is a big fan of PPAP (Pen Pineapple Apple Pen), and he has been working on finding a more convenient way to combine pens and pineapples and apples. However, those common methods are no longer enough for crazy Einsplanck! He wants to successfully combine apples and pineapples with a pen without touching fruits. After years of training, he's finally able to unleash his ultimate superpower - shooting the pen straight out of his hand!

The current situation is not good for Einsplanck. He accidentally trapped himself in a quantum realm in an experiment. After some observations, he found that the only way to escape from the quantum realm is to shoot his pen into the quantum apple and the quantum pineapple at the same time. But how can he do this?

Fortunately, after searching in his pockets, he found a plate with double slits. According to quantum mechanics, he can measure the principal quantum number n, the magnetic quantum number m, the spin quantum number s and the angular quantum number l (collectively referred to as $nmsl$) of the pen to accurately shoot it into the double-slit and split it into two pens.

However, the structures of quantum apples and quantum pineapples are very fragile. Even if he can accurately split the pen, he still needs to shoot at the exact position to hit the quantum fruits correctly. To be precise, if his shooting position is too far from the fruits, the pen will lose too much momentum before hitting the fruits, and if it is too close to the fruits, the pen's high momentum will smash them into juice.

Einsplack knows that if he is exactly k units away from both the quantum apple and the quantum pineapple, then he can use his superpowers to precisely shoot the pen into them, thereby escaping the quantum realm. Einsplack will tell you the location of the quantum fruits and the distance k. Can you help him find out how many proper shooting points there are so that he can escape from the quantum realm?

For unknown reasons (that's what the quantum realm is), all parameters about positions can only be described by integers. That is, the coordinates of the fruits and the appropriate shooting point can only be integers.

The distance Einsplack uses is the Euclidean distance, which can be calculated as the square root of the sum of the squared differences between two coordinates:

$$\text{distance}\left((x_1, y_1), (x_2, y_2)\right) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Input Format

The first line contains the number of test cases T. Then T test cases follow.

The first line of each test case contains two integers x_A and y_A, indicating the coordinates of the quantum apple. The second line contains two integers x_P and y_P, indicating the coordinates of the quantum pineapple. The third line contains a single integer k.

Output Format

Print the number of suitable shooting points.

Technical Specification

- $1 \leq T \leq 100$
- $-10^6 \leq x_A, y_A, x_P, y_P \leq 10^6$
- $0 \leq k \leq 10^6$

Sample Input 1

```
3
0 0
6 0
1
0 0
6 0
3
0 0
6 0
4
```

Sample Output 1

```
0
1
0
```

Problem B

Building a Kingdom

Time limit: 1 second

Memory limit: 2048 megabytes

Problem Description

The PCCA kingdom has n cities, numbered $1, 2, \ldots, n$. The PCCA king wants to build m bidirectional roads between the cities. Each road connects two distinct cities. From any city, one can get to any other city by moving along the roads. Furthermore, there is at most one road between any pair of cities.

However, the PCCA king finds it too costly to build all the roads, so he wants to remove some roads from the road network while building all the others. To be specific, the i^{th} road connects city u_i and city v_i, removing it costs a_i dollars, and building it costs b_i dollars. The PCCA king hopes that after building and removing some roads, any two cities can still be reached from each other by moving along the built roads.

Can you determine the minimum cost of removing and building the roads, so that the road network satisfies the king's demand?

Input Format

The first line contains two integers n and m. The i^{th} of the next m lines contains four integers u_i,v_i,a_i and b_i describing a road.

Output Format

Print one integer, the minimum total cost of removing and building the roads.

Technical Specification

- $2 \leq n \leq 2 \times 10^5$
- $1 \leq m \leq 2 \times 10^5$
- $1 \leq u_i, v_i \leq n$ for $i = 1, 2, \ldots m$
- $u_i \neq v_i$ for $i = 1, 2, \ldots m$
- $0 \leq a_i, b_i \leq 10^9$ for $i = 1, 2, \ldots m$

Sample Input 1

```
5 6
1 2 5 4
2 3 1 7
2 4 8 6
1 5 3 9
5 4 2 3
```

Sample Output 1

```
20
```

```
4 3 7 3
```

Sample Input 2

```
4 3
1 2 1000000000 1000000000
2 3 1000000000 1000000000
3 4 1000000000 1000000000
```

Sample Output 2

```
3000000000
```

Problem C
Checkerboard Splitting

Time limit: 2 seconds
Memory limit: 2048 megabytes

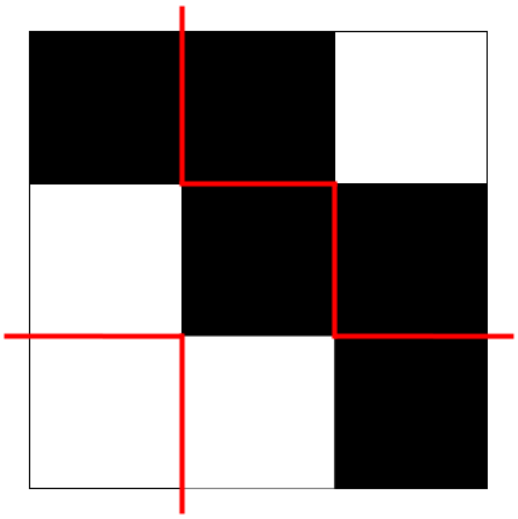
Problem Description

You are given a checkerboard with n row and m columns. The rows are numbered from 1 to n from top to bottom, and the columns are numbered from 1 to m from left to right. Each cell is identified by a pair (x, y), which means that it is located in the x^{th} row and the y^{th} column.

Initially, each cell is either black, white, or gray. You would like to paint every gray cell black or white, each with probability $\frac{1}{2}$. After that, you calculate the beauty of the board. The beauty of the board is calculated as follows:

1. First, check if there exists two positive integers i and j such that $1 \le i \le n-1$, $2 \le j \le m$, and the cells at (i, j) and $(i+1, j-1)$ have the same color. The beauty of the board is 0 if this is true.
2. Otherwise, the beauty of the board is the minimum number of the polyominos the checkerboard has to be split into, such that every pair of adjacent cells in each polyomino have different colors. A polyomino is a connected figure formed by joining one or more adjacent cells. Two cells are called adjacent if and only if they share an edge.

For example, the following checkerboard has a beauty of 3, and you can split the checkerboard into 3 polyominos by cutting along the red lines. Every pair of adjacent cells in each polyomino have different colors.



What is the expected value of the checkerboard's beauty after painting all gray cells?

Input Format

The first line contains two integers n and m. Then n lines follow, each containing m characters. The j^{th} character on the i^{th} line denotes the color of the cell at (i, j). Each character is one of

{B, W, ?} which represent the colors black, white and gray, respectively.

Output Format

Print the expected value of the checkerboard's beauty modulo 998244353.

In other words, let's write the answer as an irreducible fraction $\frac{p}{q}$, where p and q are integers and $q \not\equiv 0 \pmod{998244353}$. Print the integer x such that $0 \leq x < 998244353$ and $x \cdot q \equiv p \pmod{998244353}$.

Technical Specification

- $1 \leq n, m \leq 10^6$
- $1 \leq n \times m \leq 10^6$

Sample Input 1

```
3 5
WBWBB
WBWWW
WBBBB
```

Sample Output 1

```
4
```

Sample Input 2

```
2 2
??
??
```

Sample Output 2

```
1
```

Sample Input 3

```
3 2
?B
B?
??
```

Sample Output 3

```
0
```

Sample Input 4

```
3 7
?B????W
??????B
B??????
```

Sample Output 4

```
997025793
```

Problem D
DVD Player

Time limit: 1 second
Memory limit: 2048 megabytes

Problem Description

Darryl just bought a new television. The television's screen is w centimeters in width and h centimeters in height. We may consider a 2D Cartesian coordinate system on the screen: the bottom-left corner is $(0, 0)$, the top-right corner is (w, h), and the sides of the screen are parallel to the axes.

The television comes with a DVD player. When it is not playing, it displays a circular DVD logo of radius r centimeters that moves on the screen. At the 0^{th} second, the logo is located at (x, y) and moves at a velocity described by the vector (v_x, v_y). When the logo moves at the velocity (v'_x, v'_y) for t seconds, its x-coordinate increases by $t \cdot v'_x$ and its y-coordinate increases by $t \cdot v'_y$.

A special property of the logo is that it *bounces*, that is, its direction of movement changes whenever it touches a side of the screen:

- When it touches a vertical edge of the screen, the sign of the x component of its velocity changes.
- When it touches a horizontal edge of the screen, the sign of the y component of its velocity changes.

The following diagram shows an example of the logo bouncing off the top of the screen.

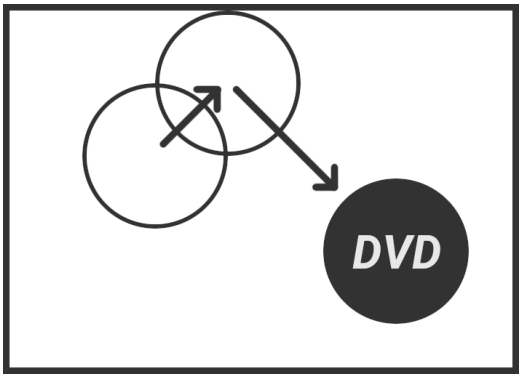


Figure 1: The logo's velocity changes from $(3, 3)$ to $(3, -3)$ after the bounce.

Darryl's dream is to watch the DVD logo touch two sides of the screen at once. Please help him determine the time at which this first happens.

Input Format

The input contains 7 integers w, h, r, x, y, v_x and v_y on a line.

Output Format

If the DVD logo will never touch two sides at once, print `-1`. Otherwise, print a real number denoting the minimum number of seconds that will have elapsed (since the 0^{th} second) when the logo touches two sides. Your answer will be accepted if the absolute or relative error is less than 10^{-6}.

Technical Specification

- $4 \leq w, h \leq 10^9$
- $1 \leq r \leq 10^8$
- $r < x < w - r$
- $r < y < h - r$
- $1 \leq v_x, v_y \leq 10^8$

Sample Input 1

```
7 5 1 2 3 3 3
```

Sample Output 1

```
1.3333333333
```

Sample Input 2

```
8 8 2 3 5 1 1
```

Sample Output 2

```
-1
```

Problem E
Elise Loves Drinks

Time limit: 1 second
Memory limit: 2048 megabytes

Problem Description

Elise loves drinks; her live isn't complete without lots of drinks.

Elise decides to buy exactly one drink everyday for the next n days. Since she likes to make plans many days ahead, she has already decided what drink to buy in the n days.

There are 10-dollar coins and 100-dollar notes in circulation. At this moment, Elise has m notes and no coins.

The price of the drink she wants to buy on the i-th day costs a_i dollars, which is a multiple of 10 and is between 10 and 90 (inclusive). Additionally, Elise buys drinks from one of the following places:

- From a convenience store: In this case, Elise can choose to pay the cashier $\frac{a_i}{10}$ coins, or to pay a note and receive $10 - \frac{a_i}{10}$ coins as change.
- From a vending machine: Since vending machines don't accept notes, Elise can only choose to pay $\frac{a_i}{10}$ coins.

Elise soon notices that it is probably impossible to pay for all the drinks using only the notes she has on the first day and the coins she receives as change during the n days. Can you help her determine if it is possible to buy all n drinks? If yes, then as Elise hates coins, she also wants to know the maximum number of notes she can keep after n days.

Input Format

The first line contains two integers n and m, the number of days to buy drinks and the number of notes Elise has.

The i^{th} of the next n lines contains two integers t_i, a_i —the place and the price of the drink to be bought on the i^{th} day. $t_i = 0$ means that the drink is from a convenience store, and $t_i = 1$ means it's from a vending machine.

Output Format

If it is impossible to buy all the drinks, print -1. Otherwise, print the maximum number of notes Elise can keep after n days.

Technical Specification

- $1 \leq n \leq 2 \times 10^5$
- $1 \leq m \leq 10^9$

- $t_i \in \{0, 1\}$ for $i = 1, 2, \ldots, n$
- $10 \leq a_i \leq 90$ for $i = 1, 2, \ldots, n$
- $10 \mid a_i$ for $i = 1, 2, \ldots, n$

Sample Input 1

```
5 12
0 90
0 10
0 30
0 20
0 40
```

Sample Output 1

```
10
```

Sample Input 2

```
5 1000000000
0 90
0 30
1 30
1 20
1 40
```

Sample Output 2

```
-1
```

Sample Input 3

```
4 3
0 90
0 90
0 90
0 90
```

Sample Output 3

```
-1
```

Sample Input 4

```
10 100
0 20
0 40
1 30
0 30
1 20
1 10
0 80
0 40
1 10
1 20
```

Sample Output 4

```
97
```

Problem F
Frog on a Line

Time limit: 1 second
Memory limit: 2048 megabytes

Problem Description

Roy is a very lazy little frog. He has no plans or dreams for his life, except jumping on a line of lotus leaves everyday.

At the beginning (the 0^{th} day), Roy is standing on the lotus at position 0 facing right. Although the little frog has no purpose in life, what he does everyday is the same. He takes exactly one of the following actions once a day:

- Turn around (change from facing left to facing right and vice versa).
- Jump to the next lotus leaf in front of him. If he is facing right, then his position increases by 1, otherwise it decreases by 1.

His friend Hong can't stand Roy's undisturbed life, and plans to visit him one day. Hong will randomly visit a certain lotus leaf on different days, and since he doesn't want Roy to see him from afar, Hong wants to approach Roy from behind to give him a big surprise.

Hong wonders which direction Roy might be facing when he's on a the lotus leaf at position x at a certain moment t. He can be facing right, facing left, facing either way, or must not be there at the moment.

Although Hong is smart, he is still a frog after all. Can you help Hong find these answers so that Hong can visit Roy?

Input Format

The first line contains the number of test cases T. Each test case contains a line with two integers t and x.

Output Format

- If it is impossible for Roy be at position x on the t^{th} day, output "`impossible`" (without quotes).
- If it is possible for Roy to be facing either direction at position x on the t^{th} day, output "`both`".
- If it is only possible for Roy to be facing right at position x on the t^{th} day, output "`right`".
- If it is only possible for Roy to be facing left at position x on the t^{th} day, output "`left`".

Technical Specification

- $1 \leq T \leq 2 \times 10^5$

- $0 \leq t \leq 10^{18}$
- $-10^{18} \leq x \leq 10^{18}$

Sample Input 1

```
4
0 0
1 0
10 4
4 10
```

Sample Output 1

```
right
left
right
impossible
```

Problem G
Glass Key

Time limit: 5 seconds

Memory limit: 2048 megabytes

Problem Description

Cinderella left the balls after dancing all night with the prince, but she left in such a rush that she dropped her glass key off! Cinderella knew the prince would try the key on every door to find her, but now she wonders if she would accidentally share the same door key with her stepsister Drizella.

After a fine amount of investigation, she found the original blueprint of Drizella's key. She noticed that blueprints of a key are unrooted tree graphs with the following properties:

- Every node has at most 3 other nodes adjacent to it.
- A node with 3 adjacent nodes is called a *joint*.
- There are at least 2 joints.
- There exist a simple path that contains every joint.

Recall that:

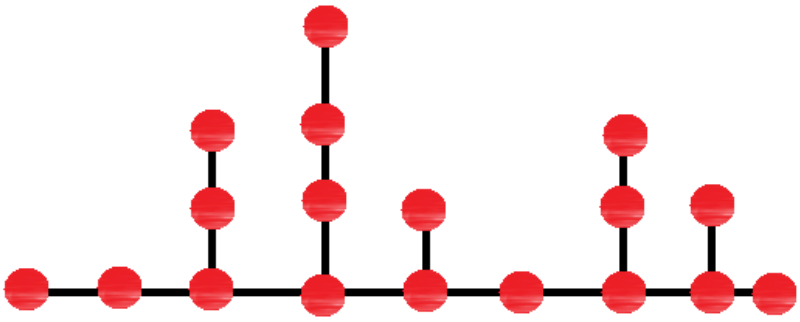
- An unrooted tree graph is a connected graph with n nodes and $n-1$ edges.
- Two nodes u and v are adjacent to each other if and only if $\{u, v\}$ is an edge in the graph.
- A simple path is a sequence $v_1, v_2, \ldots, v_k$ of vertices such that
  - v_i and v_{i+1} are adjacent for $1 \leq i \leq k-1$.
  - $v_i \neq v_j$ for $i \neq j$.

An example of a blueprint looks like this.

By reading the comments on the blueprint, she learned that a key is made from the following procedure from a blueprint:

1. The locksmith finds 2 leaf nodes a and b such that the simple path $a, u_1, u_2, \ldots, u_k, b$ contains every joint.
2. Edges on the a-b path are forged as a horizontal bar at the baseline.
3. Edges not on the path lie above the baseline as a straight vertical bars.
4. A key is described as the sequence of lengths of vertical bars above each joint, in the order from a to b.

For example, the following graph is a blueprint with the key sequence $[2, 3, 1, 2, 1]$.



Given Cinderella's key and the blueprint of Drizella's key, tell Cinderella if she can possibly share the same key with Drizella.

Input Format

The first line contains an integer n, the number of vertices in the blueprint of Drizella's key. The i^{th} of the next $n-1$ lines contains two integers a_i and b_i describing an edge in the blueprint.

The next line contains an integer m, the length of Cinderella's key. The next line contains m integers $c_1, c_2, \ldots, c_m$ describing Cinderella's key.

Output Format

If it is possible that Cinderella and Drizella share the same key, print "`YES`". Otherwise, print "`NO`".

Technical Specification

- $1 \leq m \leq n \leq 10^6$
- $0 \leq a_i, b_i \leq n-1$ for $i = 1, 2, \ldots, n$
- The edges $\{a_i, b_i\}$ form a valid key blueprint.
- $1 \leq c_j \leq 10^9$ for $j = 1, 2, \ldots, m$

Sample Input 1	Sample Output 1
14	YES

```
4 13
2 5
5 8
1 11
11 7
9 2
1 13
12 0
8 1
10 6
9 6
3 9
7 12
2
2 2
```

Almost blank page

Problem H
Hex Activation Key

Time limit: 6 seconds
Memory limit: 2048 megabytes

Problem Description

Joanna has a mysterious machine. To activate the machine, she needs to input the correct activation key. However, she doesn't know what the correct key is.

Luckily, Joanna has an old document that tells her some rules about the activation key. After reading the document, she learns that the activation key is a string that is only composed of hexadecimal symbols (`0123456789abcdef`), and the length of the correct key is n. In addition, there are m additional rules, where every rule is in one of the forms:

- `-` c_1 c_2, which means that symbol c_1 and symbol c_2 must **not** be neighboring. In other words, whenever the symbol c_1 appears, it must not be adjacent to c_2.
- `/` c_1 c_2 c_3, which means that c_1 must **not** be wrapped up by c_2 and c_3. In other words, whenever the symbol c_1 appears, it must not be adjacent to both c_2 and c_3 at the same time.

Because there may be a lot of possible activation keys, Joanna doesn't know how to find the correct one. She wonders, "How hard is it to find the correct activation key?" You don't need to find the correct activation key. Instead, you need to find the number of possible ones.

Input Format

The first line contains two integers n and m. Each of the following m lines contains the description of a rule. Each rule starts with a character t. Then,

- If $t =$ `-`, then two characters c_1 and c_2 follow.
- If $t =$ `/`, then three characters c_1, c_2, and c_3 follow.

The characters in each rule are separated by spaces.

Output Format

Print the number of possible activation keys modulo 998244353.

Technical Specification

- $1 \le n \le 10^6$
- $0 \le m \le 100$
- $t \in \{\texttt{-}, \texttt{/}\}$
- $c_1, c_2, c_3 \in \{\texttt{0}, \texttt{1}, \texttt{2}, \texttt{3}, \texttt{4}, \texttt{5}, \texttt{6}, \texttt{7}, \texttt{8}, \texttt{9}, \texttt{a}, \texttt{b}, \texttt{c}, \texttt{d}, \texttt{e}, \texttt{f}\}$

Sample Input 1

```
2 0
```

Sample Output 1

```
256
```

Sample Input 2

```
3 5
- 0 1
- a b
- 9 9
/ 4 3 5
/ 8 8 8
```

Sample Output 2

```
3938
```

Problem I
International Grandmaster

Time limit: 10 seconds

Memory limit: 2048 megabytes

Problem Description

Alice is an international chess grandmaster. Since there are no chess tournaments recently, she decides to play another game with her grandmaster friend Bob.

There is a string s and n strings $t_1, t_2, \ldots, t_n$ written on the blackboard. They take turns alternately, with Alice going first. On a player's turn, they must choose an index $i \in \{1, 2, \ldots, n\}$ and a non-empty string u that is both a suffix of t_i and a substring of s. Then, the suffix u is erased from the string t_i.

The player who can't make a move on their turn loses. Assuming both players play optimally, who has a winning strategy?

Input Format

The first line contains the string s. The next line contains the integer n. The i^{th} of the next n lines contains the string t_i.

Output Format

If Alice has a winning strategy, print "`Alice`" (without quotes). Otherwise, print "`Bob`".

Technical Specification

- $1 \leq |s| \leq 3 \times 10^5$
- $1 \leq n \leq 10^5$
- $1 \leq |t_i| \leq 3 \times 10^5$ for $i = 1, 2, \ldots, n$
- $\sum_{i=1}^{n} |t_i| \leq 3 \times 10^5$

Sample Input 1

```
ABC
3
ABC
BC
A
```

Sample Output 1

```
Bob
```

Hint

A string a is a suffix of string b if a can be obtained from b by deleting several (possibly zero) characters from the beginning.

A string a is a substring of string b if a can be obtained from b by deleting several (possibly zero) characters from the beginning and several (possibly zero) characters from the end.

Problem J
Jujutsu Kaisen

Time limit: 2.5 seconds
Memory limit: 2048 megabytes

Problem Description

Jujutsu Kaisen (呪術廻戦) is a popular Japanese anime. It's a story based on a world where all the living beings emanate an energy called "Cursed Energy", which arises from negative emotions that naturally flow throughout the body. Most people can't control Cursed Energy, so they also release it, resulting in the birth of "Curses". On the other hand, a few people can control this kind of power, who are called "Jujutsu Sorcerers". Jujutsu Sorcerers and Curses can control Cursed Energy and perform what is called Cursed Techniques.

In Jujutsu Kaisen, there is an organization named Curse Technical School which has two branches, Tokyo and Kyoto. Both branches have plenty of outstanding Jujutsu Sorcerers, with Gojo Satoru being the best among them. Each year, these two branches hold a competition called "Tokyo-Kyoto Jujutsu High School: Sister School Exchange Event" to compete with each other.

In 2022, the competition is held in team mode. There are n people in total, numbered $1, 2, \ldots, n$, and Gojo Satoru has to separate them into two teams with an equal number of people. If this is not possible, then the competition in unfair and cannot be continued. After forming the two teams, people will join 1-on-1 battles. Each person must participate in exactly one battle against a person from the other team.

To make the competition more interesting, there needs to be more negative emotions. There are m bad relationships among the n people, and Gojo Satoru hopes that every 1-on-1 battle is between two people with a bad relationship. If that happens, the competition is called a "cursed brawl".

Since Gojo Satoru is a lazy guy, he doesn't want to find a feasible way of separating people into teams such that a cursed brawl is possible. Instead, he wants to know whether for any way of separating people into two equal teams, a cursed brawl is possible.

Input Format

The first line contains two integers n and m, denoting the number of people and bad relationships. The i^{th} of the next m lines contains two integers u_i and v_i denoting a bad relationship between the people numbered u_i and v_i. Each pair of people appears at most once in the input.

Output Format

If the people cannot be separated into two teams with the same size, print one line with the string "`Not fair`" (without quotes). Otherwise, if a cursed brawl is possible for all ways of sep-

arating people into two teams, print "`Gojo Satoru`". Otherwise, print "`Ryomen Sukuna`".

Technical Specification

- $1 \leq n \leq 100$
- $0 \leq m \leq \binom{n}{2}$
- $1 \leq u_i, v_i \leq n$ for $i = 1, 2, \ldots m$
- $u_i \neq v_i$ for $i = 1, 2, \ldots m$

Sample Input 1

```
4 4
4 2
1 3
3 2
1 4
```

Sample Output 1

```
Gojo Satoru
```

Sample Input 2

```
4 2
1 3
4 2
```

Sample Output 2

```
Ryomen Sukuna
```

Sample Input 3

```
6 9
6 1
2 3
5 6
2 5
6 2
1 4
4 5
1 3
4 3
```

Sample Output 3

```
Gojo Satoru
```

Problem K
Kore wa FFT desu ka?

Time limit: 1 second
Memory limit: 2048 megabytes

Problem Description

Kumagai is a competitive programmer whose favorite algorithm is the fast Fourier transform (FFT). Whenever he encounters a problem, he asks himself the question 「これは FFT ですか？」(which means "Is this FFT?").

Today Kumagai saw a problem that has n input integers and asks for the answer modulo a prime number p. He can solve this problem with FFT if and only if there exists a positive integer x that satisfies all of the following conditions:

- x is a power of 2, that is, $x = 2^y$ for some integer y.
- $x > n$.
- x divides $p - 1$.

Please help Kumagai determine whether he can solve the problem with FFT.

Input Format

The first line contains the number of test cases T. Each test case contains a line with two integers n and p.

Output Format

For each test case, print a line containing "`YES`" (without quotes) if Kumagai can solve the problem with FFT, and "`NO`" otherwise.

Technical Specification

- $1 \leq T \leq 1000$
- $1 \leq n \leq 10^9$
- $3 \leq p < 10^9$
- p is a prime number

Sample Input 1

```
4
2 7
7 41
8 41
1234567 998244353
```

Sample Output 1

```
NO
YES
NO
YES
```