Winter Camp Contest 2022
Division 1

NYCU PCCA

ID	Problem Name	Time Limit
A	Aibohphobia	1 sec
B	Blue or Red?	1 sec
C	Checkerboard Splitting	2 sec
D	DVD Player	1 sec
E	Elise Loves Drinks	1 sec
F	Forbidden Spell	3 sec
G	Graph-Theoretic Machine	2 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
Aibohphobia

Time limit: 1 second

Memory limit: 2048 megabytes

Problem Description

Aibohphobia is the irrational fear of palindromes. A palindrome is a string that is the same forwards and backwards, for example `racecar`, `radar`, and `aibohphobia`. However, in this problem, we are not dealing with aibophobia, but instead simply solving a problem involving palindromes.

Recently, Softy participated in a algorithmic training camp, and the favorite thing she learned was Manacher's Algorithm, which can be used to efficiently compute the longest palindromic substring of a string s in $\mathcal{O}(|s|)$ time.

After burning chicken for many days in the training camp, Softy decided to procrastinate her work in the PCCA Camp for as long as possible. While wandering in the NYCU campus, she found an `std::string` stuck to the ground. Let's call the string s.

Softy was disappointed that the string isn't a palindrome, so she decided to make some changes to make it palindromic. She will repeat the following operation until s become palindromic. In the i-th turn, Softy will:

1. Choose any substring $s_l s_{l+1} \dots s_r$ $(l \le r)$ of s, let it be t.
2. Pick some intervals $[a_j, b_j]$ $(a_j \le b_j)$ such that $s_{a_j} s_{a_j+1} \dots s_{b_j}$ is equal to t and the intervals are pairwise disjoint. Note that you do not need to pick all such intervals.
3. Replace each substring $s_{a_j} s_{a_j+1} \dots s_{b_j}$ with a single character $\$_i$. You may assume that $\$_i$ is a character that never appeared in the string s before the i-th turn.

For example, let the string s be "`mississimi`". In the first turn, Softy may choose $t =$ `"mi"` and the intervals $[1, 2]$ and $[9, 10]$. After replacing $s_1 s_2$ and $s_9 s_{10}$ with $\$_1$, s will become "$\$_1$`ssissi`$\$_1$". In the second turn, Softy may choose $t =$ `"ssi"` and the intervals $[2, 4]$ and $[5, 7]$, and s will become "$\$_1\$_2\$_2\$_1$" after replacing both occurrences of `"ssi"` with $\$_2$. After this step the string becomes palindromic, so she stops.

Since the larger the palindromic string is, the happier Softy will be, she wants you to help her find out the maximum possible length of the resulting palindromic sequence. Maybe she will give you some soft Miku if you fulfill her wish?

Input Format

The input contains the string s.

Output Format

Print the maximum length of the resulting palindromic string after some operations.

Technical Specification

- $1 \le |s| \le 10^6$
- Each character in s has an ASCII code in the rage $[33, 126]$. In other words, each character is one of !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[]^_`abcdefghijklmnopqrstuvwxyz{|} .

Sample Input 1

```
mississimi
```

Sample Output 1

```
4
```

Sample Input 2

```
mirrorrorrim
```

Sample Output 2

```
12
```

Sample Input 3

```
DRUNK{NOT_THE_FLAG$1/29_10:30AM$P
CCA_final}
```

Sample Output 3

```
1
```

Problem B
Blue or Red?

Time limit: 1 second

Memory limit: 2048 megabytes

Problem Description

In the White Kingdom, there are two long-standing countries, namely Red and Blue. The two countries used to live well together, but through the TV seasons of Red vs. Blue, their relationship has grown with tensions, until finally they decided to break up. Now they are going to attract the n cities to join their party.

Prior to separation, all cities are neutral (neither red nor blue). Two different cities may be connected with a road to trade with each other. The cities are connected with m roads, in a way such that each city can be reached from any other city through the roads. During separation, each road is marked as red or blue by the White Assembly, which could not be changed anymore, but they assured us that each city is has both red and blue roads connected to it.

Each city should join either the Red country or the Blue country. The cities that belong to Red after separation are red cities, and the cities that belong to Blue are blue cities. For a separation plan to be valid, it has to satisfy both countries' conditions, all cities' conditions, and the declarations of city friendships.

The countries' conditions stated that red cities can only use red roads, while blue cities can only use blue roads. That is to say, a city needs to destroy all adjacent roads that have a different color from itself. To avoid all roads of a city being broken by neighboring cities, the countries allowed each city to pick **at least one** important red road and **at least one** important blue road adjacent to it. The country has to protect the important roads of the same color from destruction for cities that joined them (for example, Red should not allow important red roads listed by red cities to be destroyed).

The cities' conditions are simple. Some cities want to join Red, some others want to join Blue, and the rest are okay with joining either one.

Finally, we need to satisfy p pairs of relationship declarations. Some pairs of cities have declared friendly relationships with each other, while some other pairs of cities have declared unfriendly relationships. Two friendly cities want to join the same country, while two unfriendly cities want to join different countries.

We need to ensure that all conditions are satisfied to avoid rampant crimes such as crowd fighting and smuggling. Please help the two countries plan for the cities by telling each of them which country it should join, or tell them it is impossible to satisfy every condition.

Input Format

The first line contains three integers n, m and p.

The next line contains a string with n characters $t_1 t_2 \dots t_n$ denoting the cities' conditions: $t_i = \texttt{r}$ means city i wants to join Red, $t_i = \texttt{b}$ means city i wants to join Blue, and $t_i = \texttt{x}$ means city i can join either.

Each of the next m lines contains two integers a, b and three characters c, x and y describing a road connecting the cities a and b:

- If $c = \texttt{r}$ then the road is red, otherwise if $c = \texttt{b}$ then the road is blue.
- If $x = \texttt{y}$ then the road is important to a, otherwise if $x = \texttt{n}$ then it's not.
- If $y = \texttt{y}$ then the road is important to b, otherwise if $y = \texttt{n}$ then it's not.

Each of the next p lines contains a string x and two integers a and b describing a relationship declaration between cities a and b: $x = \texttt{good}$ means it's a friendly relationship, $x = \texttt{bad}$ means it's an unfriendly relationship.

Output Format

If there is no valid plan that satisfies every condition, print "`No`" (without quotes). Otherwise, print "`Yes`" on the first line, and on the second line, print a string with n characters $u_1 u_2 \dots u_n$ describing a valid plan. For each $i = 1, 2, \dots, n$, $u_i = \texttt{r}$ means city i should join Red, and $u_i = \texttt{b}$ means city i should join Blue.

Technical Specification

- $1 \le n \le 1000$
- $1 \le m \le 3000$
- $1 \le p \le \binom{n}{2}$

Sample Input 1

```
4 6 2
rxbx
1 2 r y y
3 4 b y y
1 3 r n y
1 4 b y n
2 3 b y n
2 4 r n y
good 1 2
bad 2 4
```

Sample Output 1

```
Yes
rrbb
```

Sample Input 2

```
4 4 1
xxxx
1 2 r y y
2 3 b y y
3 4 r y y
4 1 b y y
bad 1 2
```

Sample Output 2

```
No
```

Almost blank page

Problem C
Checkerboard Splitting

Time limit: 2 seconds

Memory limit: 2048 megabytes

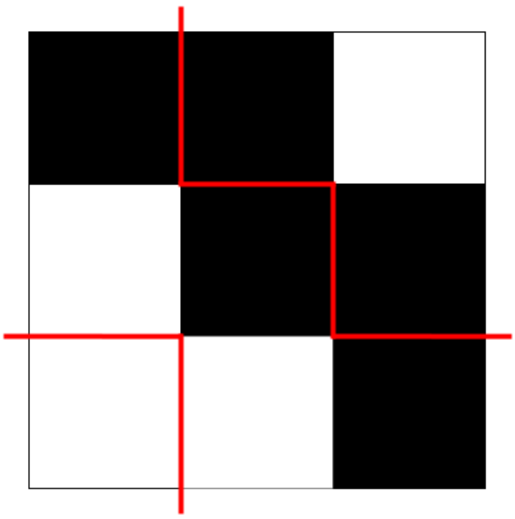
Problem Description

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

$\{\texttt{B}, \texttt{W}, \texttt{?}\}$ 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 $0 \le x < 998244353$ and $x \cdot q \equiv p \pmod{998244353}$.

Technical Specification

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

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.

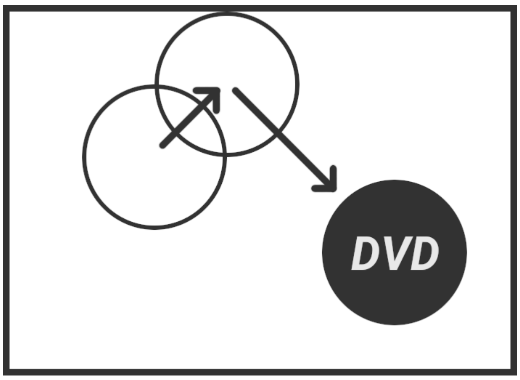


Figure 1: The logo's velocity changes from $(3, 3)$ to $(3, -3)$ after the bounce.

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

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

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

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

- $t_i \in \{0, 1\}$ 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
Forbidden Spell

Time limit: 3 seconds

Memory limit: 2048 megabytes

Problem Description

Finn is a wizard who can cast powerful spells that affect very large areas. Recently, he discovered a forbidden spell from ancient books that allows him to set a huge region on fire.

Finn lives in a country with boundaries that can be described as a convex polygon A on the Cartesian plane. The forbidden spell that he will cast affects a region inside a convex polygon B. Unfortunately, the regions inside A and B have a positive intersection area. To avoid damaging his own country, he has to cast a wind magic that changes the region B.

Finn can cast wind magic with a positive strength s, which will move the entire region B by s units along the direction parallel to the vector (d_x, d_y). Given the vector (d_x, d_y), please help Finn find the minimum strength of wind magic he has to cast to make the regions A and B no longer intersect each other. You have to answer q independent queries.

Input Format

The first line contains of an integer n. Each of the following n lines contains two integers x and y describing the coordinates of a vertex in A.

The next line contains of an integer m. Each of the following m lines contains two integers x and y describing the coordinates of a vertex in B.

The next line contains of an integer q. Each of the following q lines contains two integers d_x, d_y describe a query.

The vertices of A and B are given in counterclockwise order.

Output Format

For each query, print a real number denoting the minimum strength of the wind magic Finn has to cast. Your answer will be accepted if the absolute or relative error is less than 10^{-6}.

Technical Specification

- $3 \leq n, m \leq 2 \times 10^5$
- The coordinates of each vertex in A and B are within the range $[-10^8, 10^8]$.
- Each interior angle of A and B is less than 180 degrees.
- The intersection of the interiors of A and B has positive area.
- $1 \leq q \leq 10^5$

- $-10^8 \leq d_x, d_y \leq 10^8$
- $(d_x, d_y) \neq (0, 0)$

Sample Input 1

```
4
1 1
-1 1
-1 -1
1 -1
3
0 0
2 0
0 2
3
1 1
2 0
-2 -1
```

Sample Output 1

```
1.4142135624
1.0000000000
2.9814239700
```

Problem G
Graph-Theoretic Machine

Time limit: 2 seconds
Memory limit: 2048 megabytes

Problem Description

The PCCA kingdom has n cities numbered $1, 2, \ldots, n$. There are m bidirectional roads, each connecting two different cities. It is known that these cities and roads satisfy the following conditions:

- No two roads connect the same pair of cities.
- Each city is incident to at most 4 roads.
- It is possible to reach any city from any other city by traveling along a sequence of roads.
- If we draw a map of the kingdom, then all cities are at different locations, each road is a line segment between two cities, and no two roads intersect each other.

To attract more tourists to the kingdom, the queen has decided to color each city in one of red, green, blue or yellow. For a more exquisite view, she wants to choose the colors in a way that two cities have a different color if there is a road between them. However, finding such way of coloring is a difficult task, so she sought help from the kingdom's best scientists.

The scientists have invented the powerful Graph-Theoretic Machine (GTM). This machine works in the following way. First, the queen assigns a number to each city, called its *priority*. The priorities are integers between 1 and n (inclusive), and all cities have distinct priorities. Therefore, there are $n!$ different ways to assign priorities to all cities. Once they are assigned, the machine performs the steps below:

1. Pick an uncolored city u that satisfies the condition: if this city becomes colored, then it is possible to go from any uncolored city to any other uncolored city by traveling along roads, without visiting any colored city in between. If no such u exists, the machine crashes, and if multiple u's satisfy the condition, the one with the highest priority is chosen.
2. Pick a color c from the set $\{\text{red}, \text{green}, \text{blue}, \text{yellow}\}$ such that no city connected to u with a road is colored with c. If no such c exists, the machine crashes, and if multiple c's are valid, one of them is chosen uniformly at random.
3. Color the city u with c.
4. If there are any uncolored cities left, go back to step 1.

The queen has to assign the priorities carefully, because some assignments will lead to a chance of the machine crashing and ruining the kingdom. She wants to know the number of different ways she can assign priorities, such that the probability of the machine crashing is not zero.

Input Format

The first line contains two integer n and m, denoting the number of cities and roads. The i^{th} of the next m lines contains two integers u_i and v_i denoting a road between the cities numbered u_i and v_i.

Output Format

Print the number of different ways to assign priorities such that the machine has a nonzero probability of crashing. Output the result modulo 998244353.

Technical Specification

- $3 \leq n \leq 10^5$
- $n - 1 \leq m \leq 3n - 6$
- $1 \leq u_i, v_i \leq n$ for $i = 1, 2, \ldots m$
- $u_i \neq v_i$ for $i = 1, 2, \ldots m$

Sample Input 1

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

Sample Output 1

```
0
```

Sample Input 2

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

Sample Output 2

```
24
```

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

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

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 (呪術廻戰) 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 separating people into two teams, print "`Gojo Satoru`". Otherwise, print "`Ryomen Sukuna`".

Technical Specification

- $1 \le n \le 100$
- $0 \le m \le \binom{n}{2}$
- $1 \le u_i, v_i \le n$ for $i = 1, 2, \ldots m$
- $u_i \ne v_i$ 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
```