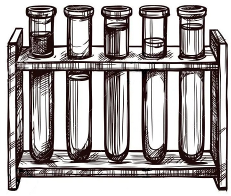
Croatian Open Competition in Informatics

Round 5, March 5th 2022

Tasks

Task	Time limit	Memory limit	Points
Kemija	1 second	512 MiB	50
Dijamant	1 second	512 MiB	70
Fliper	3 seconds	512 MiB	110
Radio	1.5 seconds	512 MiB	110
Usmjeravanje	1 second	512 MiB	110
Total			450

Task Kemija



Fran didn't pay attention in school during chemistry class and now he needs your help in doing his homework. His homework consists of n chemical equations for which he needs to check if they are balanced. A chemical equation is a way of representing chemical reactions using symbols and formulas. In a chemical reaction, some set of initial molecules react to produce a new set of molecules.

A chemical equation has a left side and a right side. The left side contains chemical formulas of the initial molecules, while the right side contains chemical formulas of the product molecules. The left and the right sides of the equation are separated by an arrow (characters `->`). Different molecules appearing on the left or the right side are separated by a `+`.

Molecules are substances made from atoms, tiny particles which are denoted with capital letters of the Latin alphabet (for the purposes of this task). The formula of a molecule is written by listing the labels of all the different atoms which make up the molecule. If a molecule has multiple instances of some atom, then the number of occurrences of this atom is written after the atom in the formula. For example, `AC4B` is a formula for a molecule which has one atom `A`, 4 atoms `C` and one atom `B`. If on one side of the equation a molecule appears more than once, then this number of occurrences is written as a coefficient in front of the formula. For example, `3AC4B` denotes 3 molecules of `AC4B`, for a total of 3 atoms `A`, 12 atoms `C` and 3 atoms `B`.

A chemical equation is said to be balanced if the right side and the left side contain an equal number of atoms of each kind. Your task is to determine whether or not each of the n chemical equations is balanced. The test cases will be such that all the numbers appearing in the reactions (the numbers of atoms in molecules and the numbers of molecules in the ractions) will have only a **single digit** (and they will be larger than 1).

Input

The first line contains a positive integer n ($1 \le n \le 10$), the number of chemical equations.

Each of the next n lines contains a sequence of characters representing a chemical equation. Each equation consists of at most 1000 characters. The equations will not necessarily be balanced, but they will be correctly written as described in the statement.

Output

For each of the n equations print `DA` if it is balanced, and `NE` if it is not, in separate lines.

Scoring

Subtask	Points	Constraints
1	10	No chemical reaction has a number in it.
2	10	No numbers appear within any formulas for molecules.
3	30	No additional constraints.

Examples

input

```
3
A+B->AB
AB+CD->AC+DB
AB+B->AB
```

output

```
DA
DA
NE
```

input

```
2
2AB+A->3AB
2AB+2AC+2BC->4ABC
```

output

```
NE
DA
```

input

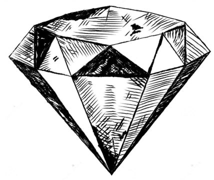
```
4
2H2O+2CO2->2H2CO3
H2SO4->H2O4
NH3+H2SO4->NH4SO4
CH4+2O2->CO2+2H2O
```

output

```
DA
NE
NE
DA
```

Clarification of the third example:
First equation: both sides have 4 atoms `H`, 2 atoms `C` and 6 atoms `O` so the answer is `DA`.
Second equation: the left side has a single `S` atom, but the right side has none so the answer is `NE`.
Third equation: the left side has 5 atoms `H`, but the right side has 4 so the answer is `NE`.
Forth equation: both sides have 4 atoms `H`, one atom `C` and 4 atoms `O` so the answer is `DA`.

Task Dijamant



Lovro has a table of n rows and m columns, where each cell is either `.` or `#`. By rotating a square by 45° a *diamond* shape is formed in the table. For a part of the table to be considered a diamond, its edge must also consist only of the character `#`, while its inside must be completely filled with `.` and it must be nonempty. Outside of a diamond any character is allowed. Diamonds come in different sizes, and the three smallest examples of a diamond are shown in the first sample.

Fabijan asked Lovro to tell him how many diamonds are there in the table, or else Lovro has to give him a cookie. Help Lovro by writing a program which counts the number of diamonds in his table.

Input

The first line contains positive integers n and m ($1 \le n, m \le 2000$), the number of rows and columns. Each of the next n lines contains m characters `.` or `#` which describe the table.

Output

In the only line print the number of diamonds in the table.

Scoring

Subtask	Points	Constraints
1	20	$1 \le n, m \le 100$
2	50	No additional constraints.

Examples

input

```
7 25
.#...#....#....#.....#...
#.#..#...#.#...#....#.#..
.#...#..#...#..#...#...#.
.....#...#.#...#..#.....#
.....#....#....#...#...#.
.....#.........#....#.#..
.....#.........#.....#...
```

output

```
3
```

input

```
11 17
.....#........#..
....#.#........#.
...#...#....#...#
..#.....#....#.#.
.#....#..#....#..
#....#.#..#......
.#....#..#.......
..#.....#........
...#...#.........
....#.#..........
.....#...........
```

output

```
1
```

input

```
5 11
##.#.#.#.##
#.#.#.#.#.#
.#.#.#.#.#.
#.#.#.#.#.#
##.#.#.#.##
```

output

```
14
```

Clarification of the second example:
There is only one diamond in the table (the one with the smallest possible size). There appears to be another diamond containing it, but it is not considered a diamond because its inside is not completely filled with '`.`'. The shape on the right side of the table is also not a diamond because it's missing a `#` character on its edge.

Task Fliper

Mirko and Slavko stumbled upon an old pinball machine. The game consists of a small metal ball that moves inside the machine, hitting and bouncing off some obstacles.

The game takes place on a board which is represented by a plane and n obstacles are given in this plane. An obstacle is represented by a line segment of unit length tilted at a 45° angle with respect to the coordinate axes. An obstacle is uniquely described by the position of its midpoint (x_i, y_i) and a character which is either `'/'` or `'\'` that denotes the orientation of the obstacle. In the plane there is also a ball which is moving in one of the four directions parallel to the axes at all times. If the ball hits an obstacle, the direction of its movement changes by 90° and the ball continues moving. Note that the obstacles are two-sided, meaning that the ball is able to bounce off both sides of the obstacle

Mirko and Slavko decided to restore the pinball machine with a new layer of paint. At their disposal are four buckets of paint containing **four different colors**. They want to paint each obstacle in one of these four colors.

Mirko: "You know, I've been thinking. There are only two possibilities for the path the ball could take."

Slavko: "What do you mean?"

Mirko: "Well, either the ball will get stuck in a cycle, periodically repeating the same sequence of bounces, or at some point it will fly off to infintiy."

Slavko: "Hm, you're right. Then maybe we could try to color the obstacles so that **for each** cycle each of the colors appears an equal number of times in the cycle and so that this number is even. For example, if some cycle consists of 24 bounces, we should make it so that there are 6 bounces for each color, and 6 is an even number."

Mirko: "And what if there is no such coloring?"

Slavko: "You know the drill, just say `-1`."

Help Mirko and Slavko determine a coloring of the obstacles which satisfies the required condition, if there is one.

Input

The first line contains a positive integer n ($1 \le n \le 500\ 000$), the number of obstacles.

The i-th of the next n lines contains integers x_i and y_i ($0 \le |x_i|, |y_i| \le 10^9$) and a character c_i (`'/'` or `'\'`) which describe the i-th obstacle. No two obstacles have the same position.

Output

If there is no coloring which satisfies the required condition, in the only line print `-1`.

Otherwise, in the only line print a sequence of n numbers `1`,`2`,`3` or `4` separated by spaces, representing a valid coloring, the colors of the obstacles in order. If there is more than one solution, print any.

Scoring

Subtask	Points	Constraints
1	20	$1 \le n \le 40$.
2	20	There is at most one cycle in which the ball can get stuck.
3	70	No additional constraints.

Examples

input

```
4
1 1 \
3 1 /
3 2 \
1 2 /
```

output

```
-1
```

input

```
9
1 2 \
1 3 /
2 1 \
2 2 \
2 3 \
3 1 /
3 2 \
4 2 /
4 3 \
```

output

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

input

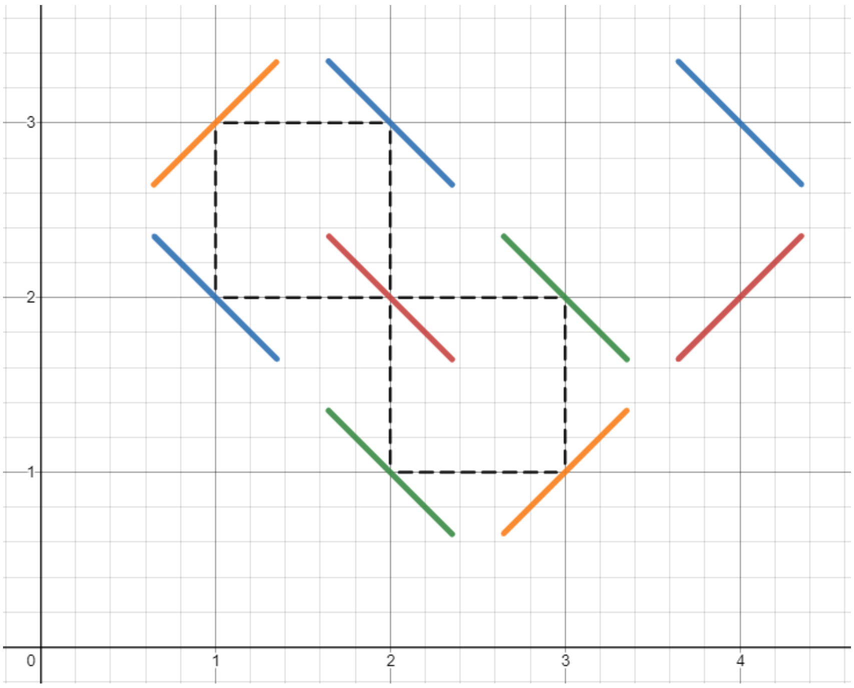
```
12
1 2 \
1 3 /
2 1 \
2 2 \
2 3 \
2 4 /
3 1 /
3 2 \
3 3 \
3 4 \
4 2 /
4 3 \
```

output

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

Clarification of the first example:
The ball can be stuck in a cycle of 4 bounces, bouncing between the 4 given obstacles. To appear an equal number of times, each color should appear once, but one is not an even number.

Clarification of the second example:
There is only one cycle in which the ball can get stuck. The image bellow shows a valid coloring. Starting from the obstacle 1, moving in the clockwise direction the colors repeat `1 3 1 4 2 3 2 4`.



Task Radio

In Croatia there are n radio stations. Each station has a concession on one of n frequencies denoted by positive integers from 1 to n. The frequencies were not chosen ideally so sometimes noise appears when multiple stations are broadcasting at the same time. To be more precise, if two radio stations, with frequencies a and b respectively, are broadcasting at the same time, noise will appear if a and b are not relatively prime. The listeners, of course, do not like the noise, so when they hear it they switch to another station.

To solve this noise problem, the radio station owners are asking you to write a program which simulates the actions of the radio stations. Your program needs to support two types of queries:

1. $S\ x$: If not currently broadcasting, the station with frequency x starts broadcasting, and if it is already broadcasting, it stops.
2. $C\ l\ r$: Check if there exists a pair of broadcasting stations whose frequencies a and b are from the interval $[l, r]$ and such that $\gcd(a, b) \neq 1$. If such a pair exists, print `DA`, otherwise print `NE`.

Initially, no station is broadcasting.

Input

The first line contains positive integers n and q ($1 \leq n \leq 1\ 000\ 000$, $1 \leq q \leq 200\ 000$), the number of radio stations (and frequencies), and the number of queries, respectively.

The i-th of the next q lines contains a description of the i-th query. For queries of the first type it will hold that $1 \leq x \leq n$, and for queries of the second type it will hold that $1 \leq l \leq r \leq n$.

Output

Print the answers to the queries of the second type in order, in separate lines.

Scoring

Subtask	Points	Constraints
1	10	$1 \leq n \leq 100$, $1 \leq q \leq 200$
2	30	For all queries of the second type it holds that $l = 1$ i $r = n$.
3	70	No additional constraints.

Examples

input

```
6 8
S 1
S 2
S 3
C 1 6
S 6
C 1 6
S 2
C 1 6
```

output

```
NE
DA
DA
```

input

```
11 6
S 4
S 10
C 3 11
C 2 7
S 6
C 2 7
```

output

```
DA
NE
DA
```

input

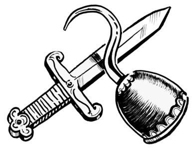
```
20 7
S 10
S 15
S 3
C 10 15
S 10
C 3 15
C 3 10
```

output

```
DA
DA
NE
```

Clarification of the first example:
The stations broadcasting during the first `C` type query are 1, 2 and 3. These numbers are all relatively prime to each other so they create no noise. Once station 6 begins to broadcast, it creates noise with stations 2 and 3. After station 2 stops broadcasting the noise with station 3 persists.

Task Usmjeravanje



Peter Pan was given career guidance to help determine his future profession. As he does not want to grow up, he ran away and sought shelter in Neverland. There are two rivers in Neverland, flowing from west to east. On the shore of the first river, there are a cities, labeled with positive integers from 1 to a in the same direction that the river flows. Similarly, on the shore of the second river there are b cities labeled in the same direction from 1 to b. Traveling downstream, it's possible to reach city j from city i if both of these cities are on the same river and if $i < j$.

Citizens of Neverland plan to establish m one-way flight routes. It is given that the i-th route should connect city x_i from the first river and city y_i from the second river, but it has not yet been decided in which direction. The citizens of Neverland would like their cities to be as connected as possible. At that moment Peter Pan realized that he would like to direct flight routes for a living.

A pair of cities is called *connected* if it is possible to reach the second city starting from the first, and vice versa. While traveling, it is allowed to use both flight routes and rivers. Peter Pan wants to determine the route directions in order to minimize the largest set of cities in which no pair of cities is connected. Help Peter Pan and determine how to direct the routes and what would the size of the mentioned set be in that case.

Input

The first line contains positive integers a, b and m ($1 \leq a, b, m \leq 200\ 000$), the number of cities on the first river, the number of cities on the second river and the number of flight routes, respectively.

The i-th of the next m lines contains two positive integers x_i and y_i ($1 \leq x_i \leq a$, $1 \leq y_i \leq b$) which denote a flight route connecting city x_i from the first river and city y_i from the second river. No pair of cities is listed more than once.

Output

In the first line print the least possible size of the maximum set of cities in which no pair of cities is connected.

In the second line print a sequence of characters `0` or `1` separated by spaces which denote the directions of the flight routes. The character `0` means that the flight takes off from the first river and lands on the second river, and conversely for `1`. If there is more than one solution, output any.

Scoring

Subtask	Points	Constraints
1	20	$1 \leq a, b, m \leq 15$
2	30	$1 \leq a, b \leq 1000$
3	60	No additional constraints.

Examples

input

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

output

```
1
1 1 0 0
```

input

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

output

```
9
1 0 1 1
```

input

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

output

```
5
1 0 1 1 0 1 0
```

Clarification of the first example:
If the flights are directed as shown in the output, it is possible to reach any city starting from any other city. Therefore, the largest set containing no pair of connected cities is a singleton. For example, it's possible to reach city 1 from the first river by starting from city 5 from the first river:

$$5\ (\text{I}) \to 3\ (\text{II}) \to 2\ (\text{I}) \to 3\ (\text{I}) \to 1\ (\text{II}) \to 2\ (\text{II}) \to 1\ (\text{I}).$$