One-Way Streets

Time Limit: 3 s   Memory Limit: 256 MB

Once upon a time there was a country with n cities and m bidirectional roads connecting them. Technical development led to faster and larger road vehicles which presented a problem—the roads were becoming too narrow for two vehicles travelling in opposite direction. A decision to solve this problem involved turning all the roads into single-lane, one-way (unidirectional) roads.

Making the roads one-way comes at a cost because some of those pairs of cities that were previously connected might no longer be reachable after the change. The government compiled a list of important pairs of cities for which it has to be possible to start in the first city and reach the second one. Your task is to determine in which direction to direct the traffic on every road. It is guaranteed that a solution exists.

For some roads there is no choice about the direction of traffic if you want to obtain a solution. The traffic will flow from the first to the second city (right direction, indicated by letter `R`) or from the second city towards the first (left direction, indicated by letter `L`). However, for some roads there exists a solution with this road directed left, and another (possibly different) solution with the road directed right. You should indicate such roads with a letter `B` for both directions.

Output a string of length m. Its i-th character should be

- `R` if all solutions require the i-th road to be directed right
- `L` if all solutions require the i-th road to be directed left
- `B` if a solution exists where the i-th road is directed left, and a solution also exists where the i-th road is directed right

Input

The first line contains the number of cities n and the number of roads m. The following m lines describe the roads with pairs of numbers a_i and b_i, which indicate that there is a road between cities a_i and b_i. There can be more than one road between the same pair of cities and a road can even connect the city with itself.

The next line contains the number of pairs of cities p that have to be reachable. The next p lines contain pairs of cities x_i and y_i, meaning that there has to be a way to start in city x_i and reach y_i.

Constraints

- $1 \leq n, m, p \leq 100\,000$
- $1 \leq a_i, b_i, x_i, y_i \leq n$

Subtask 1 (30 points)

- $n, m \leq 1000$
- $p \leq 100$

Subtask 2 (30 points)

- $p \leq 100$

Subtask 3 (40 points)

- no additional constraints

Output

Output a string of length m as described in the description of the task.

Example

Input	**Output**

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

Comment

Let's show that the fifth road "1 3" can be directed in either direction. Two possible orientations of roads with different directions of the fifth road are `LLRLRL` and `RLRRLL`.

Sure Bet

Time Limit: 2 s *Memory Limit: 128 MB*

Luck is a fundamental part of betting. Some people improve their chances and earnings by having good knowledge of what they are betting on. We will take a different approach.

Various bookmakers offer different *odds* or quotas for the same outcome. (An *odds* of x means that if you bet 1 euro and predict the outcome correctly, you get x euros back. If you predict the outcome incorrectly, you of course get nothing back. Note that you pay 1 euro regardless of the outcome.) What if you could be certain of making a profit by cleverly placing several bets? You would want to make this guaranteed profit as large as possible.

The event we want to bet on has two possible outcomes. There are n bookmakers that offer different odds. Let us denote the odds offered by the i-th bookmaker for the first outcome with a_i, and the odds offered for the second outcome with b_i. You can place a bet on any subset of the offered odds. You can even bet on both outcomes at the same bookmaker. However, all bets have to be exactly 1 euro and you cannot bet on the same outcome with the same bookmaker multiple times.

In case of the first outcome, you will receive a_i euros from every bookmaker i with whom you placed a bet on the first outcome. Similarly, in case of the second outcome, you will receive b_i euros from all eligible bookmakers. Of course in both cases, you already paid 1 euro for every bet you placed.

What is the largest *guaranteed* profit (i.e. regardless of the outcome) if you place your bets optimally?

Input

The first line contains the number of bookmakers, n. The following n lines describe the odds offered by each bookmaker with two space-separated real numbers a_i and b_i - the odds for the first and second outcome offered by the i-th bookmaker. The odds will be given to at most 4 decimal places.

Constraints

- $1.0 \leq a_i, b_i \leq 1000.0$
- $1 \leq n \leq 100\,000$

Subtask 1 (20 points)

- $n \leq 10$

Subtask 2 (40 points)

- $n \leq 1\,000$

Subtask 3 (40 points)

- no additional constraints

Output

Output the maximum guaranteed profit rounded to exactly 4 decimal places.

Here are the commands to print the floating point numbers in various languages:

- C and C++: `printf("%.4lf",(double)x);`
- Java: `System.out.printf("%.4lf",x);`
- Pascal: `writeln(x:0:4);`
- Python 3: `print("%.4lf"%x)`
- C#: `Console.WriteLine(String.Format("0:0.0000",x));`

Example

Input	Output

`4`
`1.4 3.7`
`1.2 2`
`1.6 1.4`
`1.9 1.5`
	`0.5000`

Comment

The optimal betting strategy consists of betting on the second outcome with the first bookmaker and on the first outcome with the third and fourth bookmaker. In case of the first outcome, we will earn $1.6 + 1.9 - 3 = 0.5$ and in case of the second outcome $3.7 - 3 = 0.7$. So we're guaranteed 0.5 euros regardless of the outcome.

Mousetrap

Time Limit: 5 s *Memory Limit: 512 MB*

Dumbo the elephant has a huge labyrinth with n rooms numbered $1 \ldots n$ and $n-1$ passages in such way that it is possible to reach any room from any other room. Unfortunately, a mouse sneaked into the labyrinth. Dumbo is terribly afraid of mice, so he sets a mousetrap in room t. Obviously, the mouse avoids the room with the trap, so Dumbo has to think of a better strategy to bait the mouse into the trap. The mouse constantly runs around and never stops, unless it has nowhere to move. He also knows that the mouse leaves a dirty trail of droppings and footprints in every passage it uses. The mouse then refuses to use a dirty passage again. Dumbo can clean a dirty passage, or block a passage with stones. By blocking passages or cleaning them, he wants to force the mouse to run into the trap. He would like to do this in a minimal number of moves, as he feels highly uncomfortable in the presence of a mouse.

We can describe this as a game for two players. The mouse tries to maximize the number of Dumbo's moves. The Dumbo tries to win in minimal number of moves. The first player is Dumbo. On his turn, he may clean one dirty passage of the labyrinth or block one passage. It doesn't matter if the blocked passage is clean or not. He cannot unblock a passage. He may, however, choose to do nothing. Turns in which Dumbo decides to do nothing do not count as moves. When it is the mouse's turn, the mouse will choose a clean unblocked passage and run to the adjacent room down that passage. If there is no such passage leading from mouse's current room, the mouse won't move.

Initially, all the passages are clean, the mouse is in room m, the trap is in room t, and it is Dumbo's turn. What is the minimum number of moves (passages cleaned and blocked) Dumbo needs if both players play optimally (mouse's goal is to maximize the number of Dumbo's moves)?

Input

Integers n, t and m will be given in the first line, separated by spaces. $n-1$ lines follow. In each line, a_i and b_i are given, separated by a space, which indicates a passage between rooms a_i and b_i.

Note that the input size is large.

Constraints

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

Subtask 1 (20 points)

- $n \leq 10$

Subtask 2 (25 points)

- It is guaranteed that a passage between rooms m and t exists.

Subtask 3 (20 points)

- $n \leq 1000$

Subtask 4 (35 points)

- no additional constraints

Output

Your program should print the number of Dumbo's moves.

Example

Input	Output

`10 1 4`
`1 2`
`2 3`
`2 4`
`3 9`
`3 5`
`4 7`
`4 6`
`6 8`
`7 10`
	`4`

Comment

One possible scenario:

- Dumbo blocks passage between rooms 4 and 7.
- Mouse moves to room 6. The passage between rooms 4 and 6 is now dirty.
- Dumbo blocks the passage between rooms 6 and 8.
- Mouse cannot move.
- Dumbo cleans the passage between rooms 4 and 6.
- Mouse moves to room 4. The passage between rooms 4 and 6 is dirty.
- Dumbo blocks the passage between rooms 2 and 3.
- Mouse moves to room 2. The passage between rooms 2 and 4 is dirty.
- Dumbo doesn't do anything.
- Mouse can only move to room 1 and gets caught into a trap.

Dumbo made 4 moves.