

Autonomous Driving Program Development

Input file: standard input
Output file: standard output
Time limit: 2.5 seconds
Memory limit: 1024 megabytes

Hyundai Mobis is the world's sixth largest automotive parts supplier, leading advancements in autonomous driving, infotainment, and electrification. It is reinforcing its R&D capabilities to transform into a software-centered technology company, with a focus on autonomous driving software, AI, and big data. By leveraging its hardware expertise and software innovation, Hyundai Mobis aims to strengthen its presence in the global mobility market.

The Hyundai Mobis autonomous driving testing center consists of N checkpoints. Checkpoint 1 is the starting point, and there is a junction at each checkpoint which splits to the left and right. A road may or may not be connected to each side of the junction, and if it is connected, the road can be used to travel to another checkpoint. There are exactly $N - 1$ roads and it is possible to travel to all other checkpoints from checkpoint 1. In other words, the testing center is in the form of a **binary tree whose root is checkpoint 1**.

Your objective is to write a program which moves a car at checkpoint A to checkpoint B . The program can use the following three instructions.

- L (Left turn): The car travels along the road to the left of the junction. If there is no road to the left, it raises an **error**.
- R (Right turn): The car travels along the road to the right of the junction. If there is no road to the right, it raises an **error**.
- B (Backwards): The car travels backwards along the road in the direction closer to checkpoint 1. If the car is currently at checkpoint 1, it raises an **error**.

However, the control system is broken, and the program will be executed twice as a result. For example, if the program is LRB, LRBLRB will be executed instead.

Determine whether there exists a program that, when executed twice, moves the car from checkpoint A to checkpoint B without raising an **error**. If so, find the shortest such program.

Input

The first line of input contains T , denoting the number of test cases. ($1 \leq T \leq 200\,000$)

The first line of input for each test case contains N , denoting the number of checkpoints. ($2 \leq N \leq 200\,000$)

The second line of input for each test case contains two space-separated integers A and B , denoting the starting and ending checkpoints, respectively. ($1 \leq A, B \leq N$; $A \neq B$)

Each of the following N lines of each test case contains two space-separated integers L_i and R_i .

L_i is 0 if there is no road connected to the left of the i -th checkpoint. Otherwise, it is the checkpoint number the road leads to. R_i is 0 if there is no road connected to the right of the i -th checkpoint. Otherwise, it is the checkpoint number the road leads to.

It is guaranteed that the given structure is a tree whose root is checkpoint 1.

The sum of N from all test cases is less than or equal to 1 000 000.

Output

For each test case, print the shortest program to achieve the goal. If there are multiple such programs, print any one of them. If no such program exists, print **ERROR**.

Examples

standard input	standard output
2 7 7 3 2 3 4 5 6 0 7 0 0 0 0 0 0 0 7 6 2 2 3 4 5 6 0 7 0 0 0 0 0 0 0	BBR ERROR
2 12 3 6 12 7 0 9 0 0 2 0 6 8 0 0 0 0 0 11 0 0 0 4 0 3 10 5 12 1 5 12 7 0 9 0 0 2 0 6 8 0 0 0 0 0 11 0 0 0 4 0 3 10 5	BBBBLRL ERROR