Problem F – Forecasting rock-paper-scissors

Author: Jonathan Queiroz

Roshamboland is a wonderful city, featuring an amazing riverfront, glamorous parks, and unmatched cultural attractions. Nevertheless, its inhabitants are mostly concerned with a single activity: playing rock-paper-scissors. Rock-paper-scissors is a two-player game in which each of the two players simultaneously forms one of three shapes (rock, paper or scissors) with their hands. Paper defeats rock, rock defeats scissors and scissors defeats paper. If both players form the same shape, the match ends in a draw.

A few years ago, the mayor of Roshamboland organized a huge rock-paper-scissors tournament, which was very well received by the city residents. Each day, two world-renowned players met in the Yan Ken Po Arena, the city's largest stadium, for an exciting televised match. Upon completion of each match, final results (win, loss or draw) were honorably recorded in the Official Gazette of Roshamboland.

The mayor's term is now coming to an end, and he would like to organize a new tournament as part of his (not quite legal) reelection campaign. As preparation for the upcoming tournament, he decided to hire a team of experienced Rosearchers to analyze the results of the previous edition. After months of investigation, they eventually reached the conclusion that the so-called world-renowned players always make the same moves. For example, a given player may always choose paper, while another may always choose scissors.

The mayor would like to use this groundbreaking discovery to make the next tournament more interesting. Based on logs from the Official Gazette, he needs you to figure out, for some pairs of players, who would win the match if they were to play against each other. Unfortunately the specific moves of players in the previous edition (rock, paper or scissors) were not recorded, and only match results are known (win, loss or draw). For each queried pair of players, you need to determine what would be the result of a match between them, or else, tell that there is not enough information to decide. Whenever determining the result of a match between two given players is feasible, you also need to inform the earliest moment in the tournament in which it became possible to do so. In other words, for each query involving players x and y, you also need to report the minimum integer k (if any) such that the results of the first k matches from the Official Gazette are sufficient to determine the result of a match between x and y.

Input

The first line of the input contains two integers N and M, denoting the number of players in the tournament and the number of matches that took place, respectively ($2 \leq N \leq 10^5$, $0 \leq M \leq 10^5$). Each of the next M lines describes a match, and is composed by three integers x, y and z, where x and y denote the players and z denotes the result of the match ($1 \leq x, y \leq N$, $-1 \leq z \leq 1$ and $x \neq y$). The value $z = 1$ indicates that player x won against player y, while $z = 0$ indicates a tie and $z = -1$ indicates that player x lost. It is guaranteed that match results are consistent with the rules of rock-paper-scissors, and with the fact that any given player always chooses the same shape.

The next line contains a single integer Q, indicating the number of queries that follow ($1 \leq Q \leq 3 \times 10^5$). Each of the following Q lines contains two integers x and y, denoting two players ($1 \leq x, y \leq N$, $x \neq y$).

Output

For each query between players x and y, you should print its answer in a single line. If the result of a match between the players cannot be predicted from the logs in the Official Gazette, simply print -1. Otherwise, print the earliest day in the tournament in which the match result could have been established with certainty, followed by the result itself (with 1 indicating a win for player x, 0 indicating a tie and -1 indicating a win for player y).

Sample input 1	**Sample output 1**
5 6 1 2 1 2 3 1 1 3 -1 4 5 0 1 2 1 5 4 0 6 2 3 1 3 3 5 1 2 3 2 5 4	2 1 2 -1 -1 1 1 2 -1 4 0

Sample input 2	**Sample output 2**
10 7 6 10 1 10 7 1 7 4 -1 7 5 1 4 8 -1 5 2 -1 1 3 -1 11 6 4 7 4 5 8 5 8 6 5 8 7 7 10 8 9 2 1 4 7 6 4	3 1 3 -1 5 0 5 0 4 0 5 -1 2 -1 -1 -1 3 1 3 1