Problem K. Kingdom of Jagaica

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	1024 mebibytes

The Kingdom of Jagaica is a country with n airports numbered from 1 to n. There are some airways, each of which connects two different airports bidirectionally. In other words, if an airway connects airports u and v, a passenger can move either from u to v or from v to u in a single flight. Airways may be newly established or abolished.

Mr. Threep, who is a traveler loving odd numbers, plans a trip from an airport to another one by flights. Let us say that he boards k flights: a flight from airport p_1 to airport p_2, then from p_2 to p_3, then from p_3 to p_4, and so on, and finally from p_k to p_{k+1}. This trip plan, which begins with p_1 and ends with p_{k+1}, is denoted as $p_1 \to p_2 \to p_3 \to p_4 \to \ldots \to p_k \to p_{k+1}$. According to his aesthetics, a trip plan is beautiful if each one of the n airports appears an odd number of times in this plan. For example, if $n = 6$, trip plans $3 \to 4 \to 5 \to 6 \to 1 \to 2$ and $1 \to 2 \to 3 \to 4 \to 5 \to 3 \to 2 \to 3 \to 2 \to 6$ are beautiful, while $1 \to 3 \to 6$ and $1 \to 2 \to 3 \to 4 \to 1 \to 5 \to 6$ are not. In particular, each of the n airports appears at least once in a beautiful trip plan.

Initially, there are m airways. Then, you are given q queries, which should be processed in the order they are given. Each query is of one of the two kinds below:

- "`1` x y": The existence of the airway between airports x and y changes. If there is already an airway between airports x and y, then such an airway is abolished. In other words, Mr. Threep is no longer able to board a direct flight between airports x and y (until it is newly established again). On the other hand, if there wasn't such an airway before, an airway between airports x and y is newly established. In other words, Mr. Threep can board a direct flight between airports x and y (until it is abolished again).
- "`2` x y": You have to determine whether there exists a beautiful trip plan that starts in airport x and ends in airport y using the airways that are available at this time.

Input

The first line of input contains three integers, n, m, and q: the number of airports in the kingdom, the number of airways that are initially available, and the number of queries ($2 \le n \le 10^5$; $1 \le m \le 10^5$; $1 \le q \le 10^5$).

The i-th of the following m lines contains two integers, u_i and v_i, which mean that an airway between airports u_i and v_i is initially available ($1 \le u_i < v_i \le n$). It is guaranteed that these m airways are distinct.

The j-th of the following q lines consists of three integers, t_j, x_j, and y_j, which are the type of the query and the numbers of two airports as described above ($1 \le t_j \le 2$; $1 \le x_j < y_j \le n$). It is guaranteed that there is at least one query where $t_j = 2$.

Output

For each query where $t_j = 2$, print a line with a single word: "`Yes`" if there exists a beautiful trip plan that starts in airport x and ends in airport y, or "`No`" otherwise.

Examples

standard input	*standard output*
4 2 6 1 2 3 4 2 1 2 1 2 3 2 1 2 1 2 4 1 2 3 2 1 3	No Yes Yes
5 5 4 1 2 2 3 3 4 1 4 4 5 2 1 3 2 1 4 1 2 4 2 1 4	Yes No Yes