Obliviate, Then Reincarnate

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

Leafing through his journal of musings, Mu discovered a note forgotten between the yellowed pages. The note, inspired by the faraway constellation Vulpecula, resembled a mantra of self-encouragement which read, "*Dreams are like the stars, distant and sometimes insignificant, yet enough to spark up the darkest night.*"

In time, Mu had eventually glimpsed the veiled face of dreams. Through years of growth, he had emerged more seasoned, perhaps more scarred, only to accept that stars and dreams alike, constrained by the fatalism that equally governs everything, are destined to fade into oblivion. Pursuit of dreams had crumbled, doubt about reality had shadowed, yet in his refusal to succumb to pessimism, Mu had waited, and was waiting still, harboring the belief that broken dreams, like stellar ashes in the pageantry of samsara, would be reconstructed and kindled anew.

He continued reviewing the journal, with the boundless cosmos reigning overhead. Indeed, the infinite had always held more promise and mystery than the finite — be it in the realm of spacetime, in the nebula of dreams, or even in the world of mathematics. As if on cue, his fingers paused at a page where Hilbert's Hotel Paradox, the classic paradigm of infinite sets, had once captured his contemplation. Amid the jottings, Mu had envisioned an upgraded version of the infinite hotel — the Celestial Hilbert's Hotel — and cast himself as its manager, leaving an intriguing problem for programmers like you to solve.

Hilbert's Infinite Hotel

As a marvel of deep space, the Celestial Hilbert's Hotel features infinitely many rooms. Each room is assigned a unique integer (can be positive, negative, or zero) as its number, and all the room numbers constitute $\mathbb{Z}$, the set of all integers.

There are n floors in the hotel, where rooms are distributed based on the non-negative remainder of their number when divided by n. Specifically, room x and room y are on the same floor if and only if the room numbers have the common remainder when divided by n, expressed as a congruence equation $x \equiv y \pmod{n}$.

Like any typical Hilbert's Hotel, the version in question also operates with certain relocation instructions to vacate occupied rooms. Mu has designed m relocation instructions. Each instruction, denoted as a pair (a, b), directs all guests on the same floor as room a to simultaneously move to new rooms by adding b to their previous room numbers. When necessary, Mu can select one of these m instructions and broadcast it to the corresponding floor. You might wonder about the case where guests are asked to move into already occupied rooms. In this scenario, the relocating guests will still move in, accompanying the original occupants forever.

However, due to the infinity of the Celestial Hilbert's Hotel, it is hard for Mu to track a guest's potential locations after multiple instructions, resulting in management inefficiency. Thus, Mu proposes q queries, each concerning a guest who initially occupies room x. Supposing that the relocation instructions can be applied any number of times in any order, with both the exact number of times and order being unspecified, for each query, your task is to determine whether the set of all room numbers possibly reachable by the guest concerned is an infinite set. More formally, let S_k be the set of room numbers possibly reachable by the guest concerned after at most k instructions, then you need to determine whether for all integers t, such k exists that $|S_k| > t$ can be achieved, where $|S_k|$ denotes the size of S_k.

Input

The first line contains three integers n, m, and q ($1 \leq n, m, q \leq 5 \times 10^5$), indicating the number of floors, relocation instructions, and queries, respectively.

Then m lines follow, each of which contains two integers a and b ($-10^9 \leq a, b \leq 10^9$), describing a relocation instruction (a, b).

Then q lines follow, each of which contains an integer x ($-10^9 \leq x \leq 10^9$), indicating a query concerning the guest who initially occupies room x.

Output

For each query, output "`Yes`" (without quotes) in one line if the set of all room numbers possibly reachable by the guest concerned is an infinite set, or otherwise output "`No`" (without quotes) in one line.

Examples

standard input	standard output
3 2 3 1 1 -1 3 1 2 3	Yes Yes No
3 2 3 1 1 -1 0 1 2 3	No No No