Schoolgirls

Input file:	*standard input*
Output file:	*standard output*
Time limit:	5 seconds
Memory limit:	512 mebibytes

Schoolgirl Alice learned about hinged mechanisms in her technology class. She constructed a tool that allows her to extend a parallelogram to a fourth vertex using three of its vertices (which may coincide or be collinear). Formally, given three points A, B, C, she can construct a point D such that the vectors $\overrightarrow{AB}$ and $\overrightarrow{DC}$ are equal.

Schoolgirl Alina learned about the concept of regular polygons in her geometry class. In this problem, we will use the following definitions:

- we will say that the points $A_1, A_2, \ldots, A_n$ ($n \geq 3$) *form a degenerate regular polygon* if all these points coincide;
- we will say that the points $A_1, A_2, \ldots, A_n$ ($n \geq 3$) *form a non-degenerate regular polygon in counterclockwise order* if they are all distinct, lie on the same circle with some center O, and $\angle A_1OA_2 = \angle A_2OA_3 = \ldots = \angle A_nOA_1 = \frac{360^\circ}{n} = \frac{2\pi}{n}$, and in all these angles, a counterclockwise rotation around O by $\frac{2\pi}{n}$ maps $\overrightarrow{OA_i}$ to $\overrightarrow{OA_{(i \bmod n)+1}}$;
- we will say that the points $A_1, A_2, \ldots, A_n$ ($n \geq 3$) *form a non-degenerate regular polygon* if there exists a permutation $A_{(1)}, A_{(2)}, \ldots, A_{(n)}$ of these points that forms a non-degenerate regular polygon in counterclockwise order;
- we will say that the points $A_1, A_2, \ldots, A_n$ ($n \geq 3$) *form a regular polygon* if they form a degenerate regular polygon or a non-degenerate regular polygon.

Note that the last definition is independent of the order of the points: if a list of points forms a regular polygon, then any permutation of them also forms a regular polygon.

Headmistress Arina decided to test the girls' skills. First, she gave them a task to construct $n+m$ points on the plane. The first n points should form a non-degenerate regular polygon in counterclockwise order. Each of the next m points is constructed using Alice's tool based on three previous points.

The girls coped with this part of the task. Then Arina started naming certain sets of points and asking whether they form a regular polygon. This turned out to be quite difficult for the schoolgirls, so they turned to you for help. Write a program that can handle Arina's task.

Input

The first line contains three integers n, m, k: the number of vertices in the original regular polygon, the number of additional points constructed using Alice's tool, and the number of polygons Arina will ask about ($3 \leq n \leq 10^4$, $0 \leq m \leq 3 \cdot 10^4$, $1 \leq k \leq 10^4$). The points $K_1, K_2, \ldots, K_n$ form a non-degenerate regular polygon in counterclockwise order.

The next m lines describe how the points $K_{n+1}, \ldots, K_{n+m}$ are constructed. The i-th line contains three integers a_i, b_i, c_i ($1 \leq a_i, b_i, c_i \leq n+i-1$): the numbers of the three points to which Alice's tool is applied. Point K_{n+i} is defined such that $\overrightarrow{K_{a_i}K_{b_i}} = \overrightarrow{K_{n+i}K_{c_i}}$. Some or all of the numbers a_i, b_i, c_i may coincide.

The next k lines describe Arina's sets of points. The i-th line describes the i-th set in the format "r_i $P_1^{(i)}$ $P_2^{(i)}$ $\ldots$ $P_{r_i}^{(i)}$". This means that the schoolgirls need to check whether the points $K_{P_1^{(i)}}, \ldots, K_{P_{r_i}^{(i)}}$ form a regular polygon ($3 \leq r_i \leq 3 \cdot 10^4$, $1 \leq P_j^{(i)} \leq n+m$). It is guaranteed that the sum of all r_i is at most $3 \cdot 10^4$. Some or all of the numbers $P_j^{(i)}$ may coincide.

Output

Output k lines. The i-th line should contain the word "`Yes`" if Arina's i-th set forms a regular polygon, and "`No`" otherwise. Each letter of the output can be in any case (uppercase or lowercase).

Example

standard input	*standard output*
3 6 8 1 2 3 3 1 4 5 4 3 3 1 2 4 5 3 4 5 2 6 4 7 6 5 1 2 3 1 3 2 3 1 1 8 4 2 5 6 7 3 2 1 4 3 6 5 9 3 4 7 9 4 1 3 2 8	Yes Yes Yes No No No Yes No

Note

Picture for the example: