# Redundant Towers

| | |
|---|---|
| Input file: | `standard input` |
| Output file: | `standard output` |
| Time limit: | 4 seconds |
| Memory limit: | 1024 megabytes |

There are $n$ wireless communication towers in Byteland, labeled by $1, 2, \ldots, n$. The $i$-th tower is located at $(x_i, y_i)$. No two towers share the same x-coordinate, and no two towers share the same y-coordinate. The transmission radius of each tower is $R$. In other words, the $i$-th tower can send a message to the $j$-th tower in one jump if and only if $\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \leq R$. Initially, all the towers are in use.

The $a$-th $(1 \leq a \leq n)$ tower is considered to be redundant if and only if it satisfies all the following conditions:

- The $a$-th tower is in use.
- For every pair of towers $b$ and $c$ $(1 \leq b < c \leq n, b \neq a, c \neq a)$, if they are both in use, and $b$ can send message directly or indirectly to $c$, then $b$ can also send message directly or indirectly to $c$ without passing through the $a$-th tower. Note that messages can only be transmitted by towers in use.

You are required to perform $q$ operations. In each operation, you will be given an integer $k$ $(1 \leq k \leq n)$, denoting the label of the tower whose status is changed. If the $k$-th tower is in use, it will now become not in use, and if it's not in use, it will now become in use. After each operation, you are required to count the number of redundant towers.

## Input

The first line contains two integers $n$ and $R$ $(1 \leq n \leq 10^5, 2 \leq R \leq 5)$, denoting the number of towers and the transmission radius.

In the next $n$ lines, the $i$-th line contains two integers $x_i$ and $y_i$ $(1 \leq x_i, y_i \leq n)$, denoting the location of the $i$-th tower. It is guaranteed that no two towers share the same x-coordinate, and no two towers share the same y-coordinate.

The next line contains a single integer $q$ $(1 \leq q \leq 10^5)$, denoting the number of operations.

Each of the next $q$ lines contains a single integer $k$ $(1 \leq k \leq n)$, denoting an operation. Let *last* be the previous number of redundant towers that you answered. Note that *last* should be reset to 0 at the beginning of the input. For each operation, $k$ is encrypted: its actual value is $k \oplus \mathit{last}$. In the expressions above, the symbol "$\oplus$" denotes the bitwise exclusive-or operation. Also, note that the constraints described in the statement above apply to the corresponding parameters only after decryption, the encrypted values are not subject to those constraints.

## Output

For each operation, print a single line containing a single integer, denoting the number of current redundant towers.

## Example

| standard input | standard output |
|---|---|
| 5 3<br>3 2<br>4 1<br>5 4<br>1 3<br>2 5<br>6<br>1<br>6<br>0<br>3<br>0<br>1 | 4<br>3<br>2<br>2<br>2<br>3 |