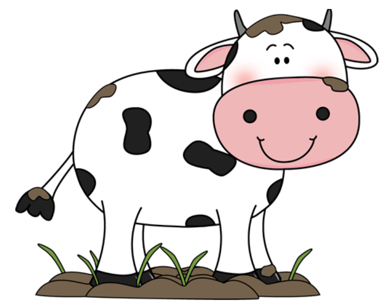
Cows



You have inherited quite a large piece of land, and upon observing it closely you realize that it is full of **cows** (and there are A LOT of them). After reflecting, you realize the cows are quite valuable and you would like to ensure they do not wander off. Luckily, the large field of land is also full of fence posts that can be used to build fences (no fence yet, but a bunch of random fence posts are scattered around). You consider whether or not you want to build a huge fence around the entire piece of land to keep the cows inside but that seems really expensive. Maybe there is a way to build a bunch of smaller fences to enclose all of the cows on your new piece of land?

Write a program that, given the 2D location of each *cow* and each *fence post*, answers queries of the following form: Given a rectangular area of the land $R = (p_1, p_2)$ (upper left corner, lower right corner), if you take all the fence posts completed contained in the area R and build a fence (straight lines between the posts), how many of the cows in region R would be enclosed in the fence and how many cows would not be enclosed in the fence?

In addition, *cows* sometimes walk around the land and change their position, so your code needs to support cows changing locations. The data stream that you have access to will let you know the new position of any cow that has changed.

Input

The first line of input is a number $5 \leq n \leq 20$. Your land stretches on the *euclidean plain* from -2^n to 2^n on both axes. All future points (cows and posts) will fall within this range.

The next line of input gives $C \leq 5 * 10^5$, the number of cows, and $P \leq 5 * 10^5$, the number of posts. The next C lines will give the x, y position of each cow and the following P lines will give the x, y locations of each post. For simplicity, all locations of cows and posts will be integers and no two cows and/or posts will be in the exact same location. It IS possible that cows and/or posts will be collinear, so keep that in mind. In addition, the points are guaranteed to be roughly uniformly distributed around the space (e.g., we won't put everything into one corner of the space).

The following line will contain the number $Q \leq 5 * 10^4$, the number of queries / updates that will be requested on the data. Each of the next Q lines will be in one of the following two formats:

- The letter Q (for query) followed by the x,y position of the upper-left corner of the range of interest, followed by the x,y position of the lower-right corner of the range of interest. These coordinates will always be integers. The sum of the total number of points that fall within these queries, over all queries, will not exceed 10^6.
- The letter M (for move) followed by an integer $i < C$, the index of the cow that has move. After this, you will be given the new x,y location of cow i.

Output

For each query, output the number of cows that are inside the given area AND inside the largest fence that can be built with the posts inside that area. After this, print the number of cows that are inside this area AND are NOT inside the largest fence that can be built with posts inside that area. If a cow falls directly on the fence line (consider how you are going to check this), then the cow is considered to be "inside" the fence. Likewise, if a cow or post fall directly on the border of the query area, then they are considered to be inside the query area.

Languages

We have written a python and java solution this time. Our python solution does not meet the time constraints so we are unsure if it is possible without significant optimizations. We highly recommend you use Java or C++ for this assignment.

Sample Input

```
10
3 5
2 1
4 1
5 2
1 0
4 0
6 2
4 5
1 5
4
Q 1 5 5 0
Q 1 5 6 0
M 2 2 2
Q 1 5 5 0
```

Sample Output

```
2 1
3 0
3 0
```