E. Predicting Popularity

time limit per test: 3 seconds
memory limit per test: 256 megabytes

Imagine that you are working at Berflix — the largest streaming service in Berland, specialized in movie distribution. The audience of this service consists of n users, and for each of them, some preferences are known: the level of action in a movie a_i and the level of drama d_i.

Your current task is to try to predict the popularity of a certain movie. Let the movie you are interested in contain ac "units" of action and dr "units" of drama (data kindly provided by the analytics team). If both the action and drama levels in the movie meet or exceed the threshold values for a certain user, they will definitely watch the movie.

If the movie falls short in either action or drama, the user will hesitate. However, the popularity of the movie among other viewers may sway them to watch it. After lengthy discussions, your team has chosen the following model of events.

Let p be the number of people who have already watched the movie (initially $p = 0$). We will consider that the movie is *suitable* for user i if $\max(a_i - ac, 0) + \max(d_i - dr, 0) \le p$.

Users constantly check recommendations. Therefore, we will assume that as long as there exists a user who has not yet watched the movie but finds it suitable, they will definitely watch it. Watching the movie will increase its popularity p by one and may make it suitable for other users.

This process will conclude when either everyone has watched the movie, or none of the remaining viewers find it suitable. Your task is to count how many people will watch the movie in total.

There is one last problem — the users' preferences are constantly changing. Specifically, there are m requests to change the values of a_k and d_k for some user k, and you need to recalculate the final popularity of the movie p after each change.

Input

The first line contains two numbers ac and dr ($1 \le ac, dr \le 10^6$) — the action and drama ratings of the movie.

The second line contains one integer n ($1 \le n \le 5 \cdot 10^5$) — the number of users of Berflix.

The third line contains n numbers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^6$) — the users' preferences for action.

The fourth line contains n numbers $d_1, d_2, \ldots, d_n$ ($1 \le d_i \le 10^6$) — the users' preferences for drama.

The fifth line contains one integer m ($1 \le m \le 3 \cdot 10^5$) — the number of changes in user preferences.

The following m lines contain the changes in the format:

- "$k_j\ na_j\ nd_j$" ($1 \le k_j \le n$; $1 \le na_j, nd_j \le 10^6$), where na_j is the new preference of user k_j for action, and nd_j — for drama.

Output

For each change request, output the total number of views of the movie p after updating the information about the corresponding user.

Example

input Copy

```
20 25
4
1 22 1 30
1 22 50 30
5
3 1 25
2 23 22
```

Educational Codeforces Round 183 (Rated for Div. 2)
Contest is running
01:34:55
Contestant

→ Submit?
Language: GNU G++20 13.2 (64 bit, w
Choose file: 选择文件 未选择文件
Submit

```
4 10 27
1 21 21
3 20 26
```

output Copy

```
3
2
4
4
0
```

Note

Consider the first request. The first and third viewers already find the movie suitable, so they will watch it, increasing the popularity p by 2. With $p = 2$, the movie will become suitable for the second viewer. As a result, they will also watch it, increasing the popularity by another 1. However, the 4-th viewer still does not find the movie suitable, as $\max(30 - 20, 0) + \max(30 - 25, 0) > 3$.

Thus, after the first request, 3 people will watch the movie.