IOI '15 P4 - Horses

Time limit: 1.0s **Memory limit:** 256M

Mansur loves to breed horses, just like his ancient ancestors did. He now has the largest herd in Kazakhstan. But this was not always the case. N years ago, Mansur was just a dzhigit (Kazakh for *a young man*) and he only had a single horse. He dreamed to make a lot of money and to finally become a bai (Kazakh for *a very rich person*).

Let us number the years from 0 to $N-1$ in chronological order (i.e., year $N-1$ is the most recent one). The weather of each year influenced the growth of the herd. For each year i, Mansur remembers a positive integer growth coefficient $X[i]$. If you started year i with h horses, you ended the year with $h \cdot X[i]$ horses in your herd.

Horses could only be sold at the end of a year. For each year i, Mansur remembers a positive integer $Y[i]$: the price for which he could sell a horse at the end of year i. After each year, it was possible to sell arbitrarily many horses, each at the same price $Y[i]$.

Mansur wonders what is the largest amount of money he could have now if he had chosen the best moments to sell his horses during the N years. You have the honor of being a guest on Mansur's toi (Kazakh for *holiday*), and he asked you to answer this question.

Mansur's memory improves throughout the evening, and so he makes a sequence of M updates. Each update will change either one of the values $X[i]$ or one of the values $Y[i]$. After each update he again asks you the largest amount of money he could have earned by selling his horses. Mansur's updates are cumulative: each of your answers should take into account all of the previous updates. Note that a single $X[i]$ or $Y[i]$ may be updated multiple times.

The actual answers to Mansur's questions can be huge. In order to avoid working with large numbers, you are only required to report the answers modulo $10^9 + 7$.

Example

Suppose that there are $N = 3$ years, with the following information:

	0	1	2
X	2	1	3
Y	3	4	1

For these initial values, Mansur can earn the most if he sells both his horses at the end of year 1. The entire process will look as follows:

- Initially, Mansur has 1 horse.
- After year 0 he will have $1 \cdot X[0] = 2$ horses.
- After year 1 he will have $2 \cdot X[1] = 2$ horses.
- He can now sell those two horses. The total profit will be $2 \cdot Y[1] = 8$.

Then, suppose that there is $M = 1$ update: changing $Y[1]$ to 2.

After the update we will have:

	0	1	2
X	2	1	3
Y	3	2	1

In this case, one of the optimal solutions is to sell one horse after year 0 and then three horses after year 2. The entire process will look as follows:

- Initially, Mansur has 1 horse.
- After year 0 he will have $1 \cdot X[0] = 2$ horses.
- He can now sell one of those horses for $Y[0] = 3$, and have one horse left.
- After year 1 he will have $1 \cdot X[1] = 1$ horse.
- After year 2 he will have $1 \cdot X[2] = 3$ horses.
- He can now sell those three horses for $3 \cdot Y[2] = 3$. The total amount of money is $3 + 3 = 6$.

Task

You are given N, X, Y, and the list of updates. Before the first update, and after every update, compute the maximal amount of money that Mansur could get for his horses, modulo $10^9 + 7$. You need to implement the functions `init`, `updateX`, and `updateY`.

```
int init(int N, int X[], int Y[])
```

- The grader will call this function first and exactly once.
- `N`: the number of years.
- `X`: an array of length N. For $0 \leq i \leq N - 1$, $X[i]$ gives the growth coefficient for year i.
- `Y`: an array of length N. For $0 \leq i \leq N - 1$, $Y[i]$ gives the price of a horse after year i.
- Note that both `X` and `Y` specify the initial values given by Mansur (before any updates).
- After init terminates, the arrays `X` and `Y` remain valid, and you may modify their contents if you wish.
- The function should return the maximal amount of money Mansur could get for these initial values of X and Y, modulo $10^9 + 7$.

```
int updateX(int pos, int val)
```

- `pos`: an integer from the range $0, \ldots, N - 1$.
- `val`: the new value for $X[\text{pos}]$.
- The function should return the maximal amount of money Mansur could get after this update, modulo $10^9 + 7$.

```
int updateY(int pos, int val)
```

- `pos`: an integer from the range $0, \ldots, N - 1$.
- `val`: the new value for $Y[\text{pos}]$.

- The function should return the maximal amount of money Mansur could get after this update, modulo $10^9 + 7$.

You may assume that all the initial, as well as updated values of $X[i]$ and $Y[i]$ are between 1 and 10^9 inclusive.

After calling `init`, the grader will call `updateX` and `updateY` an integer number of times. The total number of calls to `updateX` and `updateY` will be M.

Subtasks

subtask	points	N	M	additional constraints
1	17	$1 \leq N \leq 10$	$M = 0$	$X[i], Y[i] \leq 10$, $X[0] \cdot X[1] \cdot \ldots \cdot X[N-1] \leq 1\,000$
2	17	$1 \leq N \leq 1\,000$	$0 \leq M \leq 1\,000$	none
3	20	$1 \leq N \leq 500\,000$	$1 \leq M \leq 100\,000$	$X[i] \geq 2$ and $val \geq 2$ for `init` and `updateX` respectively
4	23	$1 \leq N \leq 500\,000$	$1 \leq M \leq 10\,000$	none
5	23	$1 \leq N \leq 500\,000$	$1 \leq M \leq 100\,000$	none