Коњи

Mansur сака да одгледува коњи. Сега има најголемо стадо во селото. Но, кога бил млад бил копук со само еден коњ. Тој сонувал да се збогати и да стане чорбаџија.

Нека ги нумерираме годините со 0 до $N-1$ во хронолошки редослед (т.е., годината $N-1$ е најпоследната). Временските услови секоја година се различни и влијаат на порастот на стадото. За секоја година i, Mansur памти позитивен цел број како коефициент за раст $X[i]$. Ако годината започнала i со h коњи во стадото, истата ќе заврши со $h \cdot X[i]$ во стадото.

Коњите може да се продаваат на крајот на годината. За секоја година i, Mansur памти позитивен цел број $Y[i]$: цената за која можел да продаде еден коњ, на крај на годината i. После секоја година, можело да се продадат произволно многу коњи, секој по цена од $Y[i]$.

Mansur се прашува која е најголемата сума пари која тој ќе можел да ја заработи ако ги изберел најдобрите моменти за продажба на коњите во текот на N-те години. ВИе сте на гости кај Mansur и тој вам ви го поставил прашањето.

Сеќавањата му се враќаат на Mansur во текот на вечерта, па затоа прави низа од M ажурирања. Секое ажурирање ќе промени или една вредност од $X[i]$ или една вредност од $Y[i]$. После секое ажурирање, тој пак препрашува за најголемата сума која би ја имал со продажба на коњите. Ажурирањата на Mansur се кумулативни: секој ваш одговор треба да ги земе предвид сите претходни ажурирања. Запазете дека едно исто $X[i]$ или $Y[i]$ може да се ажурира повеќе пати.

Одговорот кој Mansur може да го добие може да е ОГРОМЕН. За да се избегне работа со големи броеви вие треба да го дадете одговорот ПО МОДУЛ $10^9 + 7$.

Пример

Нека има $N = 3$ години, со следната информација:

	0	**1**	**2**
X	2	1	3
Y	3	4	1

За овие почетни вредности, Mansur може да заработи најмногу ако ги продаде двата коња после годината 1. Целиот процес ќе оди вака:

- Прво, Mansur има 1 коњ.
- После годината 0 ќе има $1 \cdot X[0] = 2$ коњи.
- После годината 1 ќе има $2 \cdot X[1] = 2$ коњи.

- Сега може да продаде 2 коња. Вкупно ќе заработи $2 \cdot Y[1] = 8$.

Потоа, претпоставете дека има $M = 1$ ажурирање: промена на $Y[1]$ во 2.

По ажурирањето имаме:

	0	**1**	**2**
X	2	1	3
Y	3	2	1

Во овој случај едно оптимално решение е да продаде еден коњ после годината 0 и потоа 3 коња после годината 2.
Целиот процес ќе оди вака:

- Прво, Mansur има 1 коњ.
- После годината 0 ќе има $1 \cdot X[0] = 2$ коња.
- Сега може да продаде еден од тие коњи за цена $Y[0] = 3$, и за му остане еден коњ.
- После годината 1 ќе има $1 \cdot X[1] = 1$ коњ.
- После годината 2 ќе има $1 \cdot X[2] = 3$ коњи.
- Сега сите 3 може да ги продаде по цена $3 \cdot Y[2] = 3$. Вкупната заработка е $3 + 3 = 6$.

Задача

Дадени ви се N, X, Y, и листа на ажурирања. Пред првото ажурирање и после секое ажурирање, пресметајте ја максималната сума на пари која Mansur може да ја добие за своите коњи, ПО МОДУЛ $10^9 + 7$.

You need to implement the functions `init`, `updateX`, and `updateY`.

- `init(N, X, 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 $10^9 + 7$.
- `updateX(pos, val)`

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

- `updateY(pos, val)`
  - `pos`: an integer from the range $0, \dots, N-1$.
  - `val`: the new value for $Y[\texttt{pos}]$.
  - 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` several times. The total number of calls to `updateX` and `updateY` will be M.

Subtasks

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

Sample grader

The sample grader reads the input from the file `horses.in` in the following format:

- line 1: `N`
- line 2: `X[0]` … `X[N - 1]`
- line 3: `Y[0]` … `Y[N - 1]`
- line 4: `M`
- lines 5, …, `M` + 4: three numbers `type pos val` (`type=1` for `updateX` and `type=2` for `updateY`).

The sample grader prints the return value of `init` followed by the return values of all calls to `updateX` and `updateY`.