Horses

Мансур обича да отглежда коне, както своите древни прародители. Сега той има голямо стадо в Казахстан. Но това не винаги е било така. Преди N години, когато Мансур беше dzhigit (в превод от казахски *млад мъж*), той имаше само един кон. Той мечтаеше да спечели много пари и накрая да стане bai (в превод от казахски *много богат човек*).

Нека да номерираме годините от 0 до $N-1$ в хронологичен ред (т.е. година $N-1$ е последната). Времето през всяка от годините влияеше върху нарастването на стадото. За всяка година i Мансур помни положителен цял коефициент на растежа $X[i]$. Ако започва i-тата година с h коня, той приключва годината с $h \cdot X[i]$ коня в стадото си.

Конете могат да се продават само в края на годината. За всяка година i Мансур знае положително цяло число $Y[i]$: цената, за която той може да продаде един кон в края на i-тата година. След всяка година е възможно да се продадат произволен брой коне, всеки на цена $Y[i]$.

Мансур се пита каква е най-голямата обща сума, която може да има сега, ако избира най-добрите моменти за продажба на конете през N-те години. За вас е чест да гостувате на Мансур и той ви моли да му отговорите на този въпрос.

Паметта на Мансур се подобрява през цялата вечер. Затова той прави редица от M обновявания. Всяко обновяване променя или една от стойностите на $X[i]$, или една от стойностите на $Y[i]$. След всяко обновяване той отново ви пита каква е най-голямата обща сума, която може да спечели от продажбата на конете си. Обновяванията на Мансур се натрупват: всеки от вашите отговори трябва да отчита всички предишни обновявания. Забележете, че някои $X[i]$ или $Y[i]$ може да се обновяват и повече от веднъж.

Отговорите на въпросите на Мансур може да са много големи. За да се избегне работата с дълги числа, от вас се иска да дадете отговора по модул $10^9 + 7$.

Пример

Нека $N = 3$ години и имаме следната информация:

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

При тези първоначални стойности Мансур ще спечели най-много, ако продаде двата си коня в края на година 1.
Целият процес ще изглежда по следния начин:

- В началото Мансур има 1 кон.
- В края на година 0 той ще има $1 \cdot X[0] = 2$ коня.
- В края на година 1 той ще има $2 \cdot X[1] = 2$ коня.
- Сега той може да продаде тези два коня. Общата печалба ще бъде $2 \cdot Y[1] = 8$.

Да предположим, че след това има $M = 1$ обновяване: променяме $Y[1]$ на 2.

След обновяването ще имаме:

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

В този случай едно от оптималните решения е да се продаде един кон в края на година 0 и след това три коня в края на година 2.
Целият процес ще изглежда по следния начин:

- Първоначално Мансур има един кон.
- В края на година 0 той ще има $1 \cdot X[0] = 2$ коня.
- Сега той може да продаде един от тези коне за $Y[0] = 3$, след което ще му остане един кон.
- В края на година 1 той ще има $1 \cdot X[1] = 1$ кон.
- В края на година 2 той ще има $1 \cdot X[2] = 3$ коня.
- Сега той може да продаде тези три коня за $3 \cdot Y[2] = 3$. Общата му печалба от продажбите ще е $3 + 3 = 6$.

Задача

Дадени са N, X, Y, както и списък от обновявания. Преди първото обновяване и след всяко обновяване трябва да пресметнете максималната обща сума пари, която Мансур може да получи за своите коне, по модул $10^9 + 7$.
Трябва да имплементирате функции `init`, `updateX` и `updateY`.

- `init(N, X, Y)` — Грейдърът първо ще извика тази функция. Функцията ще бъде извикана само веднъж.
  - `N`: броят на годините.
  - `X`: масив с дължина N. За $0 \leq i \leq N - 1$, $X[i]$ задава коефициента на растеж за i -тата година.
  - `Y`: масив с дължина N. За $0 \leq i \leq N - 1$, $Y[i]$ задава цената на един кон в края на i-тата година.
  - Обърнете внимание, че стойностите и на двата масива `X` и `Y` съответстват на началните стойности, описани от Мансур преди каквито и да било обновявания.

- След като `init` приключи, масивите X и Y остават валидни и може да променяте тяхното съдържание, ако желаете.
- Функцията трябва да върне максималната обща сума пари, които Мансур може да получи при тези начални стойности на X и Y, по модул 10^9+7.

- `updateX(pos, val)`
  - `pos`: цяло число в интервала $0, \ldots, N-1$.
  - `val`: нова стойност за $X[\text{pos}]$.
  - Функцията трябва да върне максималната обща сума пари, които Мансур може да получи след това обновяване, по модул 10^9+7.
- `updateY(pos, val)`
  - `pos`: цяло число в интервала $0, \ldots, N-1$.
  - `val`: новата стойност за $Y[\text{pos}]$.
  - Функцията трябва да върне максималната обща сума пари, които Мансур може да получи след това обновяване, по модул 10^9+7.

Може да считате, че всички стойности на $X[i]$ и $Y[i]$ както в началото, така и след всяко обновяване са между 1 и 10^9 включително.
След извикването на `init` грейдърът ще извиква `updateX` и `updateY` няколко пъти. Общият брой на извикванията на `updateX` и `updateY` ще бъде M.

Подзадачи

подзадача	точки	N	M	допълнителни ограничения
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	$1 \le N \le 1,000$	$0 \le M \le 1,000$	няма
3	20	$1 \le N \le 500,000$	$0 \le M \le 100,000$	$X[i] \ge 2$ and $val \ge 2$ за `init` и `updateX` съответно
4	23	$1 \le N \le 500,000$	$0 \le M \le 10,000$	няма
5	23	$1 \le N \le 500,000$	$0 \le M \le 100,000$	няма

Примерен грейдър

Примерният грейдър чете входните данни от файл `horses.in` в следния формат:

- ред 1: `N`
- ред 2: `X[0]` … `X[N - 1]`
- ред 3: `Y[0]` … `Y[N - 1]`
- ред 4: `M`

- редове 5, …, $M + 4$: три числа `type pos val` (`type=1` за `updateX` и `type=2` за `updateY`).

Примерният грейдър отпечатва върнатата стойност от `init`, последвана от върнатите стойности от всички извиквания на `updateX` и `updateY`.