Horses

Lui Mansur îi place să crească cai urmând tradiția strămoșilor săi. El are acum ce mai mare herghelie din Kazakhstan. Nu așa stăteau lucrurile cu N ani în urmă. Când Mansur era doar un dzhigit (cuvântul Kazakh pentru *tânăr*) el avea doar un singur cal. El visa să facă o grămadă de bani și în cele din urmă să ajungă un bai (cuvântul kazakh pentru *om foarte bogat*).

Să numerotăm anii de la 0 la $N-1$ în ordine cronologică (adică anul $N-1$ este cel mai recent an). Clima din fiecare an influența creșterea hergheliei. Pentru fiecare an i Mansur memorează un coeficient de creștere întreg și pozitiv $X[i]$. Dacă la începutul anului i aveai h cai atunci la sfârșitul acestuia aveai $h \cdot X[i]$ cai în herghelie.

Caii puteau fi vânduți numai la sfârșitul unui an. Pentru fiecare an i, Mansur memorează un întreg pozitiv $Y[i]$: prețul unui cal la sfârșitul anului i. La sfârșitul fiecărui an era posibil să vinzi oricâți cai, fiecare la acelasi preț $Y[i]$.

Mansur se întreabă care este cea mai mare sumă de bani pe care ar putea să o obțină dacă alege cele mai bune momente în care să vândă cai pe parcursul celor N ani. Tu ai onoarea să fii invitaltul lui Mansur în toi (cuvântul kazakh pentru *vacanță*) și să răspunzi la întrebarea lui.

Memoria lui Mansur se îmbunătățește seara, așa ca va face un șir de M modificări. Fiecare modificare va schimba fie una dintre valorile $X[i]$, fie una dintre valorile $Y[i]$. După fiecare modificare el te întreabă dinnou care e suma cea mai mare pe care o poate obține din vânzarea cailor. Modificările lui Manur sunt cumulative: fiecare răspuns trebuie să țină cont de toate modificările precedente. Rețineți că oricare dintre valorile $X[i]$ sau $Y[i]$ ar putea fi modificată de mai multe ori.

Răspunsul lui Mansur poate fi un număr foarte mare. Pentru a evita lucrul cu numere mari se cere doar restul modulo 10^9+7 al răspunsului.

Exemplu

Să presupunem că $N = 3$ ani, cu următoarele informații:

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

Pentru valorile inițiale Mansur poate obține cel mai mult dacă vinde ambii săi cai la sfârșitul anului 1. Procesul decurge după cum urmează:

- Inițial, Mansur are un cal.
- După anul 0 el are $1 \cdot X[0] = 2$ cai .

- După anul 1 el are $2 \cdot X[1] = 2$ cai .
- El poate acum să vândă cei doi cai. Profitul total va fi $2 \cdot Y[1] = 8$.

Să presupunem acum că există $M = 1$ modificări: Schimbă valoarea lui $Y[1]$ în 2.

După modificare avem:

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

În acest caz, una dintre soluțiile optime este să vinzi un cal după anul 0 și apoi trei cai după anul 2. Procesul decurge după cum urmează:

- Inițial, Mansur are un cal.
- După anul 0 el are $1 \cdot X[0] = 2$ cai.
- El poate să vândă unul dintre cai pentru $Y[0] = 3$, și îi mai rămâne un cal.
- După anul 1 el are $1 \cdot X[1] = 1$ cal.
- După anul 2 el are $1 \cdot X[2] = 3$ cai.
- El poate acum să vândă cei trei cai pentru $3 \cdot Y[2] = 3$. Profitul total va fi $3 + 3 = 6$.

Cerință

Se dau N, X, Y, și lista de modificări. Înainte de prima modificare și după fiecare modificare, calculează suma maximă pe care o poate obține Mansur pe caii săi, modulo $10^9 + 7$.
Trebuie să implementezi funcțiile `init`, `updateX` și `updateY`.

- `init(N, X, Y)` — Grader-ul va apela prima această funcție, exact o dată.
  - `N`: Numărul de ani.
  - `X`: un șir de lungime N. Pentru $0 \le i \le N-1$, $X[i]$ dă coeficientul de creștere pentru anul i.
  - `Y`: un șir de lungime N. Pentru $0 \le i \le N-1$, $Y[i]$ dă prețul unui cal după anul i.
  - Remarcați că atât `X` cât și `Y` specifică valorile inițiale date de Mansur (înainte de orice modificare).
  - După ce apelul `init` se încheie, șirurile `X` și `Y` rămân valabile, și poți modifica conținutul lor după cum dorești.
  - Această funcție trebuie să returneze suma maximă pe care o poate obține Mansur pe caii săi pentru aceste valori inițiale ale lui X și Y, modulo $10^9 + 7$.
- `updateX(pos, val)`
  - `pos`: un întreg din intervalul $0, \ldots, N-1$.

- `val`: noua valoare a lui $X[\text{pos}]$.
- Această funcție trebuie să returneze suma maximă pe care o poate obține Mansur după această modificare, modulo $10^9 + 7$.

- `updateY(pos, val)`
  - `pos`: un întreg din intervalul $0, \ldots, N - 1$.
  - `val`: noua valoare a lui $Y[\text{pos}]$.
  - Această funcție trebuie să returneze suma maximă pe care o poate obține Mansur după această modificare, modulo $10^9 + 7$.

Se asigură că atât valorile inițiale cât și cele modificate pentru $X[i]$ și $Y[i]$ sunt între 1 și 10^9 inclusiv.
După `init`, grader-ul va apela `updateX` și `updateY` de câteva ori. Numărul total de apeluri ale funcțiilor `updateX` și `updateY` va fi M.

Subprobleme

Subproblema	puncte	N	M	Precizări suplimentare
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	$1 \le N \le 500,000$	$0 \le M \le 100,000$	$X[i] \ge 2$ și $val \ge 2$ pentru `init` și apelurile `updateX`
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

Grader-ul de pe calculatorul tău

Grader-ul de pe calculatorul tău citește date de intrare din fișierul `horses.in` în următorul format:

- linia 1: `N`
- linia 2: `X[0]` … `X[N - 1]`
- linia 3: `Y[0]` … `Y[N - 1]`
- linia 4: `M`
- liniile 5, …, `M` + 4: trei numere `type pos val` (`type=1` pentru `updateX` și `type=2` pentru `updateY`).

Grader-ul de pe calculatorul tău afișează valoarea returnată de apelul `init` urmată de valorile returnate de toate apelurile `updateX` și `updateY`.