Cavalls

A en Mansur li agrada molt criar cavalls, de la mateixa manera que als seus avantpassats. Avui dia té el ramat mes gran de Kazakhstan. Però això no sempre ha estat així. Fa N anys, en Mansur només era un dzhigit (que en kazakh vol dir *un home jove*) i tenia un sol cavall. Sempre somiava amb fer un munt de diners i esdevenir un bai (que en kazakh vol dir *una persona molt rica*).

Numerem els anys de 0 a $N-1$ en ordre cronològic (és a dir, l'any $N-1$ és el més recent). El clima de cada any afectava al creixemnt del ramat. Per cada any i, en Mansur recorda el coeficient de creixement $X[i]$, que és un enter positiu. Si començava l'any i amb h cavalls, acabava l'any amb $h \cdot X[i]$ cavalls en el seu ramat.

Els cavalls només es podien vendre al final de cada any. Per cada any i, en Mansur recorda un enter positiu $Y[i]$: el preu pel qual podia vendre el cavall al final de l'any i. Després de cada any, és possible vendre un nombre qualsevol de cavalls, tots al mateix preu $Y[i]$.

En Mansur es pregunta quina és la quantitat de diners que podria tenir ara si hagués escollit els millors moments per vendre els seus cavalls durant els N anys. Tens el gran honor de ser un convidat a les toi (*vacances* en kazakh) d'en Mansur, i t'ha demanat la resposta a aquesta pregunta.

La memòria d'en Mansur va millorant durant la tarda, i per tant va fent una seqüencia d'actualitzacions. Cada actualització canviarà un dels valors $X[i]$ o un dels valors $Y[i]$. Després de cada actualització torna a preguntar per la màxima quantitat que podria haver obtingut venent els seus cavalls. Les actualitzacions d'en Mansur són acumulatives: cadascuna de les respostes ha de tenir en compte totes les actualitzacions prèvies. Fixeu-vos que un mateix valor $X[i]$ o $Y[i]$ pot ser actualitzat en més d'una ocasió.

Les respostes exactes a les preguntes d'en Mansur poden ser enormes. Per tal d'evitar treballar amb enters grans, se't demana que diguis les respostes mòdul 10^9+7.

Exemple

Suposeu que hi ha $N=3$ anys, amb la següent informació:

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

Per a aquests valors inicials, en Mansur pot obtenir el màxim si ven els seus dos cavalls després de l'any 1. El procés sencer serà:

- Inicialment, en Mansur té 1 cavall.
- Després de l'any 0 tindrà $1 \cdot X[0] = 2$ cavalls.

- Després de l'any 1 tindrà $2 \cdot X[1] = 2$ cavalls.
- Ara pot vendre els dos cavalls. El benefici total serà $2 \cdot Y[1] = 8$.

Llavors, suposeu que hi ha $M = 1$ actualització: canviar $Y[1]$ a 2.

Després de l'actualització tindrem:

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

En aquest cas, una solució òptima és vendre un cavall després de l'any 0 i llavors 3 cavalls després de l'any 2. El procés sencer serà el següent:

- Inicialment, en Mansur té 1 cavall.
- Després de l'any 0 tindrà $1 \cdot X[0] = 2$ cavalls.
- Ara pot vendre un dels seus cavalls per $Y[0] = 3$, i quedar-se amb un cavall.
- Després de l'any 1 tindrà $1 \cdot X[1] = 1$ cavall.
- Després de l'any 2 tindrà $1 \cdot X[2] = 3$ cavalls.
- Ara pot vendre els 3 cavalls per $3 \cdot Y[2] = 3$. El total de diners serà $3 + 3 = 6$.

El problema

Se us donen N, X, Y, i la llista d'actualitzacions. Abans de la primera actualització, i després de cadascuna d'elles, calculeu la quantitat màxima de diners que en Mansur podria aconseguir pels seus cavalls, mòdul $10^9 + 7$. Se us demana que implementeu les funcions `init`, `updateX` i `updateY`.

- `init(N, X, Y)` — El grader cridarà aquesta funció en primer lloc i una vegada exactament.
  - `N`: el nombre d'anys.
  - `X`: un array de mida N. Per cada $0 \leq i \leq N-1$, $X[i]$ dóna el coeficient de creixement de l'any i.
  - `Y`: un array de mida N. Per cada $0 \leq i \leq N-1$, $Y[i]$ dóna el preu d'un cavall un cop acaba l'any i.
  - Fixeu-vos que tant `X` com `Y` especifiquen els valors inicials donats per en Mansur (abans de cap actualització).
  - Després que `init` acabi, els arrays `X` i `Y` continuen sent vàlids i podeu modificar-ne el contingut si voleu.
  - La funció haurà de retornar la màxima quantitat de diners que en Mansur podria obtenir per aquests valors inicials de X i Y, mòdul $10^9 + 7$.
- `updateX(pos, val)`
  - `pos`: un enter contingut a l'interval $0, \ldots, N-1$.

- `val`: el nou valor de $X[\text{pos}]$.
- La funció haurà de retornar la quantitat màxima de diners que en Mansur podria obtenir després d'aquesta actualització, mòdul $10^9 + 7$.

- `updateY(pos, val)`
  - `pos`: un enter contingut a l'interval $0, \ldots, N - 1$.
  - `val`: el nou valor de $Y[\text{pos}]$.
  - La funció haurà de retornar la quantitat màxima de diners que en Mansur podria obtenir després d'aquesta actualització, mòdul $10^9 + 7$.

Podeu suposar que tots els valors tant inicials com actualitzats de $X[i]$ i $Y[i]$ es troben entre 1 i 10^9 inclusivament.
Després de cridar la funció `init`, el grader cridarà `updateX` i `updateY` diverses vegades.
Anomenarem M al nombre total de crides a `updateX` i `updateY`.

Subtasques

subtasca	punts	N	M	restricions addicionals
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	$1 \leq N \leq 1{,}000$	$0 \leq M \leq 1{,}000$	cap
3	20	$1 \leq N \leq 500{,}000$	$0 \leq M \leq 100{,}000$	$X[i] \geq 2$ i $val \geq 2$ a les crides a `init` i `updateX` respectivament
4	23	$1 \leq N \leq 500{,}000$	$0 \leq M \leq 10{,}000$	cap
5	23	$1 \leq N \leq 500{,}000$	$0 \leq M \leq 100{,}000$	cap

Sample grader

El sample grader llegeix l'entrada de l'arxiu `horses.in` en el format següent:

- línia 1: `N`
- línia 2: `X[0]` … `X[N - 1]`
- línia 3: `Y[0]` … `Y[N - 1]`
- línia 4: `M`
- línies 5, …, `M` + 4: tres nombres `type pos val` (`type=1` per a `updateX` i `type=2` per a `updateY`).

El sample grader imprimeix el valor que retorna `init` seguit dels valors que retornen totes les crides a `updateX` i `updateY`.