Horses

Mansur is net als zijn voorvaderen een paardenfokker. Hij heeft nu de grootste kudde in Kazkahstan. Dat was echter niet altijd zo. N jaar geleden was Mansur een dzhigit (Kazakhstaans voor *een kereltje*) en had hij maar één paard. Zijn droom was om veel geld te verdienen en een bai te worden (Kazakhstaans voor *een rijke vent*).

Nummer de jaren in chronologische volgorde van 0 tot en met $N-1$ (dus $N-1$ is het meest recente jaar). Het weer is elk jaar van invloed op de omvang van de kudde. Voor elk jaar i herinnert Mansur een positieve integer groeicoefficient $X[i]$. Als hij aan het begin van jaar i precies h paarden had, dan heeft hij aan het einde van het jaar $h \cdot X[i]$ paarden.

Paarden kunnen alleen aan het einde van een jaar verkocht worden. Voor elk jaar i herinnert Mansur een positieve integer $Y[i]$: het bedrag waarvoor hij een paard aan het einde van het jaar kon verkopen. Aan het eind van elk jaar was het mogelijk om een willekeurig aantal paarden te verkopen, elk tegen het tarief van $Y[i]$.

Mansur vraagt zich af wat het grootste geldbedrag is dat hij nu zou kunnen hebben als hij de beste momenten had gekozen om paarden te verkopen gedurende de N jaar. Omdat je een bijzondere gast bent op Mansur's toi (Kazakhstaans voor *vakantie*) heeft hij jou gevraagd om deze vraag te beantwoorden.

Het geheugen van Mansur wordt gedurende de avond beter. Daarom geeft hij je M updates. Elke update verandert ofwel een $X[i]$ ofwel een $Y[i]$. Na elke update vraagt Mansur je opnieuw wat de grootste hoeveelheid geld is die hij verdiend zou kunnen hebben door paarden te verkopen. Mansur's updates zijn cumulatief: elk antwoord moet rekening houden met alle voorgaande updates. Let op dat een enkele $X[i]$ of $Y[i]$ meerdere keren upgedate mag worden.

De antwoorden op de vragen van Mansur kunnen heel erg groot worden. Om te vermijden met grote getallen te werken moet je je antwoorden geven modulo 10^9+7.

Voorbeeld

Stel dat er $N = 3$ jaren zijn met de volgende informatie:

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

Voor deze initiele waarden geldt dat Mansur het meeste geld verdient als hij beide paarden verkoopt aan het einde van jaar 1. Het hele proces ziet er als volgt uit:

- Mansur begint met 1 paardje.

- Na jaar 0 heeft hij $1 \cdot X[0] = 2$ paardjes.
- Na jaar 1 heeft hij $2 \cdot X[1] = 2$ paardjes.
- Hij kan nu beide paardjes verkopen. Zijn totale opbrengst is $2 \cdot Y[1] = 8$.

Stel dat er vervolgens $M = 1$ update is: verander $Y[1]$ naar 2.

Na de update ziet de tabel er als volgt uit:

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

Nu is een van de optimale oplossingen om 1 paardje te verkopen na jaar 0 en vervolgens drie paardjes na jaar 2. Het hele proces ziet er als volgt uit:

- Mansur begint met 1 paardje.
- Na jaar 0 heeft hij $1 \cdot X[0] = 2$ paardjes.
- Hij verkoopt een van de paardjes voor $Y[0] = 3$ en heeft dan nog 1 paardje over.
- Na jaar 1 heeft hij $1 \cdot X[1] = 1$ paardjes.
- Na jaar 2 heeft hij $1 \cdot X[2] = 3$ paardjes.
- Hij kan deze drie paardjes verkopen voor $3 \cdot Y[2] = 3$. Zijn totale inkomsten zijn $3 + 3 = 6$.

Opdracht

Je krijgt N, X, Y en de lijst met updates. Voor de eerste update, en na elke update, moet je berekenen wat de maximale inkomsten van Mansur hadden kunnen zijn, module $10^9 + 7$. Je moet de functies `init`, `updateX` en `updateY` implementeren.

- `init(N, X, Y)` — De grader roept deze functie als eerste aan, precies één keer.
  - `N`: het aantal jaren.
  - `X`: een array van lengte N. Voor alle $0 \leq i \leq N-1$, geeft $X[i]$ de groeicoefficient voor jaar i.
  - `Y`: een array van lengte N. Voor alle $0 \leq i \leq N-1$, geeft $Y[i]$ de prijs van een paardje aan het einde van jaar i.
  - Merk op dat `X` en `Y` ook de initiele waarden van Mansur geven (dus voor updates).
  - Wanneer `init` klaar is dan blijven de arrays `X` en `Y` bestaan; je mag hun waardes zelf aanpassen als je dat wil.
  - De functie moet als resultaat de maximale opbrengst retourneren die Mansur had kunnen behalen voor de initiele waarden van X en Y, modulo $10^9 + 7$.
- `updateX(pos, val)`

- `pos`: een integer in het bereik $0, \ldots, N-1$.
- `val`: de nieuwe waarde voor $X[\text{pos}]$.
- De functie moet als resultaat de maximale opbrengst retourneren die Mansur had kunnen behalen na deze update, modulo $10^9 + 7$.

- `updateY(pos, val)`
  - `pos`: een integer in het bereik $0, \ldots, N-1$.
  - `val`: de nieuwe waarde voor $Y[\text{pos}]$.
  - De functie moet als resultaat de maximale opbrengst retourneren die Mansur had kunnen behalen na deze update, modulo $10^9 + 7$.

Je kunt ervan uitgaan dat alle initiele waarden, en alle upgedate waarden van $X[i]$ en $Y[i]$ een waarde van 1 tot en met 10^9 zijn.
Na de aanroep van `init` zal de grader verschillende keren `updateX` en `updateY` aanroepen. Het totaal aantal aanroepen van `updateX` en `updateY` noemen we M.

Subtasks

subtask	punten	N	M	aanvullende voorwaarden
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	$1 \leq N \leq 1{,}000$	$0 \leq M \leq 1{,}000$	geen
3	20	$1 \leq N \leq 500{,}000$	$0 \leq M \leq 100{,}000$	$X[i] \geq 2$ danwel $val \geq 2$ voor `init` en `updateX`
4	23	$1 \leq N \leq 500{,}000$	$0 \leq M \leq 10{,}000$	geen
5	23	$1 \leq N \leq 500{,}000$	$0 \leq M \leq 100{,}000$	geen

Voorbeeld grader

De voorbeeld grader leest de invoer uit het bestand `horses.in` in het volgende formaat:

- regel 1: `N`
- regel 2: `X[0]` … `X[N - 1]`
- regel 3: `Y[0]` … `Y[N - 1]`
- regel 4: `M`
- regel 5, …, `M` + 4: drie getallen `type pos val` (`type=1` voor `updateX` en `type=2` voor `updateY`).

De voorbeeld grader drukt de return waarde van `init` af, gevolgd door de return waardes van alle aanroepen van `updateX` en `updateY`.