Koně

Mansúr stejně jako jeho předkové rád chová koně a má v současnosti největší stádo v Kazachstánu. Nicméně ne vždy tomu tak bylo. Před N lety byl Mansúr pouhý dzhigit, což v kazaštině znamená *mladík*, a měl pouze jediného koně. Snil o tom vydělat hodně peněz a stát se bai, což značí *velmi bohatého člověka*.

Označme roky v chronologickém pořadí od 0 do $N-1$, kde $N-1$ je rok současný. Růst stáda v každém roce závisí na počasí. Pro každý rok i si Mansúr pamatuje kladné celé číslo $X[i]$, které představuje koeficient růstu stáda. Má-li stádo na začátku roku i počet koní h, bude mít Mansúr na konci tohoto roku $h * X[i]$ koní ve stádu.

Mansúr může prodávat koně pouze vždy na konci roku. Pro každý rok i si pamatuje kladné celé číslo $Y[i]$ udávající cenu, za níž může na konci roku i prodat jednoho koně. Na konci roku může prodat libovolný počet koní, a to všechny za stejnou cenu, tj. každého za $Y[i]$.

Mansúr by rád znal maximální částku peněz, které by mohl mít, kdyby koně během N let prodával v nejlepší chvíli. Máte to potěšení být Mansúrovým prázdninovým hostem a můžete mu pomoci tuto otázku zodpovědět.

Jelikož se Mansúrovi jeho paměť s průběhem večera zlepšuje, postupně provádí M aktualizací, kdy v každé z nich buďto změní jednu hodnotu $X[i]$, nebo jednu hodnotu $Y[i]$. Po každé aktualizaci se Mansúr znovu ptá na maximální částku peněz, které mohl vydělat prodejem svých koní. Aktualizace jsou kumulativní, tzn. každá vaše odpověď musí vzít v úvahu všechny dosavadní změny. Jednotlivé hodnoty $X[i]$ nebo $Y[i]$ mohou být změněny i vícekrát.

Skutečné odpovědi na Mansúrovy otázky mohou být obrovské. Abyste se vyhnuli počítání s velkými čísly, částku v odpovědi uveďte modulo $10^9 + 7$.

Příklad

Předpokládejme, že máme $N = 3$ roky s následujícími údaji:

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

Pro tyto počáteční hodnoty Mansúr nejvíc vydělá, prodá-li koně na konci prvého roku. Celý proces bude vypadat následovně:

- Mansúr má na začátku 1 koně.
- Po roce 0 bude mít $1 \cdot X[0] = 2$ koně.

- Po roce 1 bude mít $2 \cdot X[1] = 2$ koně.
- Nyní tyto dva koně může prodat. Celkový příjem z prodeje bude $2 \cdot Y[1] = 8$.

Nyní předpokládejme $M = 1$ aktualizaci: změnu $Y[1]$ na novou hodnotu 2.

Po aktualizaci máme:

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

V tomto případě je jedním z optimálních řešení prodat jednoho koně po roce 0 a následně tři koně po roce 2.
Celý proces bude vypadat takto:

- Na začátku má Mansúr jednoho koně.
- Po roce 0 bude mít $1 \cdot X[0] = 2$ koně.
- Nyní může jednoho z nich prodat za $Y[0] = 3$ a jeden mu zůstane.
- Po roce 1 bude mít $1 \cdot X[1] = 1$ koně.
- Po roce 2 bude mít $1 \cdot X[2] = 3$ koně.
- Nyní může tyto tři koně prodat za $3 \cdot Y[2] = 3$. Celkem takto získá $3 + 3 = 6$ peněz.

Úloha

Máte dány hodnoty N, X, Y a posloupnost aktualizací. Před první aktualizací a po každé další spočtěte maximum získatelných peněz za koně modulo $10^9 + 7$.
Implementujte funkce `init`, `updateX` a `updateY`.

- `init(N, X, Y)` — Vyhodnocovač ji zavolá jako první a to právě jednou.
  - `N`: počet let.
  - `X`: pole délky N. Pro $0 \leq i \leq N - 1$, $X[i]$ udává koeficient růstu stáda v daném roce.
  - `Y`: pole délky N. Pro $0 \leq i \leq N - 1$, $Y[i]$ udává cenu koně po daném roce.
  - Uvědomte si, že `X` a `Y` představují počáteční hodnoty dané Mansúrem ještě před aktualizacemi.
  - Funkce vrátí maximální částku peněz, jež může Mansúr získat pro tyto inicální hodnoty X a Y, modulo $10^9 + 7$.
- `updateX(pos, val)`
  - `pos`: celé číslo z rozsahu $0, \ldots, N - 1$.
  - `val`: nová hodnota pro $X[pos]$.
  - Funkce vrátí maximální částku peněz, jež může Mansúr získat po této aktualizaci, modulo

$10^9 + 7$.

- `updateY(pos, val)`
  - `pos`: celé číslo z rozsahu $0, \ldots, N-1$.
  - `val`: nová hodnota pro $Y[pos]$.
  - Funkce vrátí maximální částku peněz, jež může Mansúr získat po této aktualizaci, modulo $10^9 + 7$.

Můžete předpokládat, že jak počáteční, tak aktualizované hodnoty $X[i]$ a $Y[i]$ jsou mezi 1 a 10^9 včetně.
Po zavolání `init` vyhodnocovač několikrát zavolá `updateX` a `updateY`. Celkový počet volání `updateX` a `updateY` bude M.

Podúlohy

podúloha	bodů	N	M	další omezení
1	17	$1 \le N \le 10$	$M = 0$	$X[i], Y[i] \le 10$, $X[0] \cdot X[1] \cdot \ldots \cdot X[N-1] \le 1000$
2	17	$1 \le N \le 1000$	$0 \le M \le 1000$	žádná
3	20	$1 \le N \le 500000$	$0 \le M \le 100000$	$X[i] \ge 2$, resp. $val \ge 2$ pro `init`, resp. `updateX`
4	23	$1 \le N \le 500000$	$0 \le M \le 10000$	žádná
5	23	$1 \le N \le 500000$	$0 \le M \le 100000$	žádná

Ukázkový vyhodnocovač

Ukázkový vyhodnocovač čte vstup ze souboru `horses.in` v následujícím tvaru:

- řádek 1: `N`
- řádek 2: `X[0]` … `X[N - 1]`
- řádek 3: `Y[0]` … `Y[N - 1]`
- řádek 4: `M`
- řádky 5, …, `M` + 4: tři čísla `type pos val` (`type=1` pro `updateX` a `type=2` pro `updateY`).

Ukázkový vyhodnocovač vypíše návratovou hodnotu `init` následovanou návratovými hodnotami všech volání `updateX` a `updateY`.