

Horses

Veleposjednik Mansur voli da gaji konje, baš kao što su to činili i njegovi preci. On sada posjeduje najveće krdo u Kazahstanuu. No, to nije oduvijek bio slučaj. Prije N godina, Mansur je bio običan džigit (kazahstanska riječ za *onaj mladi bilmez bez ičega*) i imao je svega jednog konja. Sanjao je o tome da zaradi mnogo para i konačno postane baja (kazahstanska riječ za *čovjek sa zlatnom kajlom*).

Označimo godine od 0 do $N - 1$, hronološki (tj. godina $N - 1$ je najskorija). Vremenski uslovi su svake godine uticali na porast Mansurovog krda. Za svaku godinu i , Mansur zna pozitivan cio broj $X[i]$ koji predstavlja stopu rasta krda. Ako je godinu i počeo sa h konja (h je cio broj), tu je godinu završio sa krdom veličine $h \cdot X[i]$ konja.

Na kraju svake godine Mansur je mogao prodati određeni broj konja. Za svaku godinu i , Mansur zna pozitivan cio broj $Y[i]$: cijenu za koju je mogao prodati jednog konja na kraju te godine. Nakon svake godine mogao je prodati proizvoljan broj konja, svakog po cijeni $Y[i]$.

Mansur se pita koliko bi najviše para do sada mogao zaraditi da je odabrao najbolje momente da proda svoje konje u toku prethodnih N godina. Imali ste čast da vas Mansur pozove na toi (kazahstanska riječ za slavu ili *prekomjerno uživanje u različitim porocima*), i tom prilikom je od vas zatražio da odgovorite na njegovo pitanje.

Dok dejstvo poroka slabi, Mansurovo sjećanje postaje sve bolje, i on pravi M izmjena u podacima koje je naveo. Svaka izmjena mijenja ili jednu od $X[i]$ vrijednosti ili jednu od $Y[i]$ vrijednosti. Nakon svake izmjene on vas iznova pita koliko je najviše para mogao da zaradi prodavajući svoje konje. Mansurove izmjene su kumulativne: svaki vaš odgovor treba da uzme u obzir sve prethodne izmjene. Svako $X[i]$ ili $Y[i]$ može biti izmijenjeno više puta.

Tačan odgovor na Mansurovo pitanje može biti ogroman. Da biste izbjegli rad sa velikim brojevima, od vas se samo traži da saopštite odgovor po modulu $10^9 + 7$.

Primjer

Neka je $N = 3$, sa sljedećim vrijednostima za X i Y :

	0	1	2
X	2	1	3
Y	3	4	1

Za ove početne vrijednosti, Mansur može zaraditi najviše ako proda oba svoja konja na kraju godine 1. Cio proces bi izgledao ovako:

- Na početku Mansur ima jednog konja.

- Nakon godine 0 će imati $1 \cdot X[0] = 2$ konja.
- Nakon godine 1 će imati $2 \cdot X[1] = 2$ konja.
- On sada može prodati ta 2 konja. Ukupna zarada će biti $2 \cdot Y[1] = 8$.

Zatim, pretpostavimo da imamo $M = 1$ izmjenu: mijenjamo $Y[1]$ u 2.

Nakon ove izmjene X i Y su:

	0	1	2
X	2	1	3
Y	3	2	1

U ovom slučaju, jedno od optimalnih rešenja je da Mansur proda jednog konja na kraju godine 0 a zatim 3 konja na kraju godine 2. Cio proces bi izgledao ovako:

- Na početku Mansur ima jednog konja.
- Nakon godine 0 imaće $1 \cdot X[0] = 2$ konja.
- On sada može prodati jednog od njih za $Y[0] = 3$, i preostaće mu 1 konj.
- Nakon godine 1 imaće $1 \cdot X[1] = 1$ konja.
- Nakon godine 2 imaće $1 \cdot X[2] = 3$ konja.
- On sada može prodati ta 3 konja za $3 \cdot Y[2] = 3$, što čini ukupnu zaradu od $3 + 3 = 6$.

Zadatak

Date su vam vrijednosti N , X , Y i lista izmjena. Prije prve izmjene, i nakon svake sljedeće izmjene, izračunajte najveću moguću zaradu koju Mansur može imati od prodaje svojih konja po modulu $10^9 + 7$. Treba da implementirate funkcije `init`, `updateX` i `updateY`.

- `init(N, X, Y)` — Ocjenjivač će pozvati prvo ovu funkciju i pozvaće je tačno jednom.
 - N : ukupan broj godina.
 - X : niz dužine N . Za $0 \leq i \leq N - 1$, $X[i]$ predstavlja koliko puta se umnožio broj konja u godini i .
 - Y : niz dužine N . Za $0 \leq i \leq N - 1$, $Y[i]$ predstavlja cijenu jednog konja na kraju godine i .
 - Primjetite da X i Y predstavljaju početne Mansurove vrijednosti (prije bilo kakvih izmjena).
 - Nakon što se funkcija `init` završi, nizovi X i Y ostaju važeći i možete mijenjati vrijednosti u njima po želji.
 - Funkcija treba da vrati vrijednost koja predstavlja najveću moguću zaradu koju Mansur može da ostvari za početne vrijednosti nizova X i Y , po modulu $10^9 + 7$.
- `updateX(pos, val)`

- pos: cio broj iz intervala $0, \dots, N - 1$.
 - val: nova vrijednost za $X[pos]$.
 - Funkcija treba da vrati vrijednost koja predstavlja najveću moguću zaradu koju Mansur može da ostvari nakon ove izmjene, po modulu $10^9 + 7$.
- updateY(pos, val)
- pos: cio broj iz intervala $0, \dots, N - 1$.
 - val: nova vrijednost za $Y[pos]$.
 - Funkcija treba da vrati vrijednost koja predstavlja najveću moguću zaradu koju Mansur može da ostvari nakon ove izmjene, po modulu $10^9 + 7$.

Možete pretpostaviti da su sve početne vrijednosti, kao i izmijenjene vrijednosti $X[i]$ i $Y[i]$ između 1 i 10^9 , uključivo.

Nakon pozivanja funkcije `init`, ocjenjivač će pozivati funkcije `updateX` i `updateY` više puta. Ukupan broj poziva funkcija `updateX` i `updateY` je M .

Podzadaci

podzadatak	bodovi	N	M	dodatna ograničenja
1	17	$1 \leq N \leq 10$	$M = 0$	$X[i], Y[i] \leq 10$, $X[0] \cdot X[1] \cdot \dots \cdot X[N - 1] \leq 1,000$
2	17	$1 \leq N \leq 1,000$	$0 \leq M \leq 1,000$	nema
3	20	$1 \leq N \leq 500,000$	$0 \leq M \leq 100,000$	$X[i] \geq 2$ i $val \geq 2$ za funkcije <code>init</code> i <code>updateX</code> respektivno
4	23	$1 \leq N \leq 500,000$	$0 \leq M \leq 10,000$	nema
5	23	$1 \leq N \leq 500,000$	$0 \leq M \leq 100,000$	nema

Ocjenjivač (grader)

Ocjenjivač čita ulazne podatke iz fajla `horses.in` u sljedećem formatu:

- red 1: N
- red 2: $X[0] \dots X[N - 1]$
- red 3: $Y[0] \dots Y[N - 1]$
- red 4: M
- red 5, ..., $M + 4$: tri broja `type pos val` (`type=1` za `updateX` i `type=2` za `updateY`).

Ocjenjivač prvo štampa vrijednost koju vraća funkcija `init` a zatim i povratne vrijednosti funkcija `updateX` i `updateY` u redosljedu u kome se pozivaju.