Konji

Mansur voli uzgajati konje, baš kako su to radili njegovi pretci. On trenutno posjeduje najveće krdo u Kazahstanu. Ali nije uvijek bilo tako. Prije N godina, Mansur je bio obični dzhigit (Kazačka riječ za *mlad čovjek*) i imao je samo jednog konja. Sanjao je kako će se obogatiti i konačno postati bai (Kazačka riječ za *vrlo bogata osoba*).

Označimo godine brojevima od 0 do $N-1$ u kronološkom poretku (odnosno, godina $N-1$ je zadnja). Vremenski uvjeti utjecali su na rast krda svake godine. Za svaku godinu i, Mansur se sjeća koeficijenta rasta $X[i]$ koji je prirodan broj. Ako godinu i počnemo s h konja, na kraju godine imat ćemo $h \cdot X[i]$ konja u krdu.

Konji se mogu prodavati samo na kraju godine. Za svaku godinu i, Mansur se sjeća prirodnog broja $Y[i]$: cijene za koju je mogao prodati jednog konja na kraju godine i. Na kraju svake godine moguće je prodati proizvoljan broj konja, svakog po istoj cijeni $Y[i]$.

Mansur se pita koja je najveća svota novca koju bi sada imao da je kroz svih N godina birao najbolje trenutke za prodaju svojih konja. Imate čast biti gost na Mansurovu toi (Kazački za *blagdan*), i on vas je zamolio da mu odgovorite na pitanje.

Mansur se kroz večer prisjeća sve točnijih podataka, i tako radi niz od M ispravaka. Svaki ispravak mijenja jednu od vrijednosti $X[i]$ ili jednu od vrijednosti $Y[i]$. Nakon svakog ispravka opet vas pita za najveću količinu novca koju bi zaradio prodajom konja. Mansurovi ispravci su kumulativni: svaki odgovor mora uzeti u obzir sve prethodne ispravke. Primijetite da isti $X[i]$ ili $Y[i]$ može biti ispravljen više puta.

Točni odgovori na Mansurova pitanja mogu biti velika. Kako bismo izbjegli rad s velikim brojevima, odgovore morate vratiti modulo 10^9+7.

Primjer

Pretpostavite da je $N=3$, sa sljedećim informacijama:

	0	1	2
X	2	1	3
Y	3	4	1

Za ove početne vrijednosti, Mansur može zaraditi najviše ako proda oba konja na kraju prve godine. Cijeli proces izgleda ovako:

- U početku Mansur ima jednog konja.
- Nakon godine 0 imat će $1 \cdot X[0] = 2$ konja.

- Nakon godine 2 imat će $2 \cdot X[1] = 2$ konja.
- Tada može prodati ta dva konja za konačni profit od $2 \cdot Y[1] = 8$.

Nadalje, pretpostavimo da postoji $M = 1$ ispravaka: promjena $Y[1]$ na 2.

Nakon ispravka imamo:

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

U ovom slučaju, jedno od optimalnih rješenja je prodati jednog konja nakon godine 0, i zatim tri konja nakon godine 2.
Cijeli proces izgleda ovako:

- U početku Mansur ima jednog konja.
- Nakon godine 0 imat će $1 \cdot X[0] = 2$ konja.
- Sada može prodati jednog od tih konja za $Y[0] = 3$, nakon čega mu ostaje samo jedan.
- Nakon godine 1 imat će $1 \cdot X[1] = 1$ konja.
- Nakon godine 2 imat će $1 \cdot X[2] = 3$ konja.
- Sada može prodati ta tri konja za $3 \cdot Y[2] = 3$. Ukupna količina novca je $3 + 3 = 6$.

Zadatak

Zadani su N, X, Y, i lista ispravaka. Prije prvog ispravka i nakon svakog ispravka, izračunajte maksimalnu količinu novca koju Mansur može dobiti za svoje konje modulo $10^9 + 7$.
Morate implementirati funkcije `init`, `updateX` and `updateY`.

- `init(N, X, Y)` — Grader će pozvati ovu funkciju točno jednom.
  - `N`: broj godina.
  - `X`: niz duljine N. Za $0 \leq i \leq N-1$, $X[i]$ daje koeficijent rasta za svaku godinu.
  - `Y`: niz duljine N. Za $0 \leq i \leq N-1$, $Y[i]$ daje cijenu konja nakon svake godine.
  - Primijetite da `X` i `Y` odgovaraju početnim vrijednostima koje vam Mansur daje (prije bilokakvih ispravaka).
  - Funkcija mora vratiti maksimalnu količinu novca koju Mansur može dobiti za početne vrijednosti X i Y, modulo $10^9 + 7$.
- `updateX(pos, val)`
  - `pos`: cijeli broj iz $0, \ldots, N-1$.
  - `val`: nova vrijednost za $X[pos]$.
  - Funkcija mora vratiti maksimalnu količinu novca koju Mansur može dobiti nakon ispravka,

modulo $10^9 + 7$.

- `updateY(pos, val)`
  - `pos`: cijeli broj iz $0, \ldots, N-1$.
  - `val`: nova vrijednost za $Y[pos]$.
  - Funkcija mora vratiti maksimalnu količinu novca koju Mansur može dobiti nakon ispravka, modulo $10^9 + 7$.

Možete pretpostaviti da početne, kao i ispravljene vrijednosti $X[i]$ i $Y[i]$ leže u intervalu 1 do 10^9 uključivo.
Nakon poziva funkcije `init`, grader će pozivati `updateX` i `updateY` više puta.
Ukupni broj poziva `updateX` i `updateY` će biti M.

Podzadaci

podzadatak	bodovi	N	M	dodatna ograničenja
1	17	$1 \leq N \leq 10$	$M = 0$	$X[i], Y[i] \leq 10$, $X[0] \cdot X[1] \cdot \ldots \cdot X[N-1] \leq 1,000$
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 `init` i `updateX` redom
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

Lokalni grader

Lokalni grader čita ulaz iz datoteke `horses.in` u sljedećem formatu:

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

Lokalni grader ispisuje povratnu vrijednost poziva funkcije `init`, zatim povratne vrijednosti poziva funkcija `updateX` i `updateY`.