



Cavalli

Mansur, come i suoi antenati, ha fatto carriera allevando e vendendo cavalli e oggi la sua mandria è la più numerosa di tutto il Kazakistan. Non sempre però le cose stavano così: N anni fa, Mansur era un semplice dzhigit (*ragazzo*) e aveva soltanto un cavallo. Sognava però di guadagnare molti soldi e diventare finalmente un bai (*uomo benestante*).

Numeriamo gli anni da 0 a $N - 1$ in ordine cronologico (dove l'anno $N - 1$ è il più recente). Il clima dei vari anni ha influenzato la crescita della mandria: per ogni anno i , Mansur ricorda il *coefficiente di crescita* $X[i]$ (intero positivo) della mandria. Questo coefficiente indica che, se all'inizio dell'anno i nella mandria ci sono h cavalli, alla fine dell'anno questo numero diventerà $h \cdot X[i]$.

La vendita dei cavalli avviene solo alla fine dell'anno, e per ogni anno i Mansur ricorda un intero positivo $Y[i]$: il prezzo per il quale era possibile vendere un cavallo alla fine dell'anno i . Durante una vendita si può decidere di vendere un numero qualsiasi di cavalli, ciascuno allo stesso prezzo $Y[i]$.

Mansur ora si chiede qual è la più grande somma di denaro che potrebbe avere se avesse scelto il momento migliore in cui vendere i cavalli, nel corso degli N anni. Dal momento che ti è stato concesso l'onore di essere ospite nella tua (*festa*) di Mansur, ora devi aiutarlo a trovare la risposta alla sua domanda.

La memoria di Mansur migliora con il passare della serata, nel corso della quale ti propone M correzioni. Ogni correzione va a modificare un valore $X[i]$ oppure un valore $Y[i]$. Dopo ciascuna correzione, Mansur ti chiede di nuovo qual è la più grande somma di denaro che avrebbe potuto guadagnare vendendo i suoi cavalli. Le correzioni fatte sono cumulative: ogni tua risposta deve tenere in conto tutte le correzioni precedenti.

Nota: uno stesso $X[i]$ o $Y[i]$ potrebbe subire più correzioni nel corso della serata.

Le risposte alle domande di Mansur potrebbero essere enormi. Per evitare di gestire numeri molto grandi, ti viene richiesto soltanto di stampare la risposta modulo $10^9 + 7$.

Esempio

Supponiamo che ci siano $N = 3$ anni e che Mansur ricordi le seguenti informazioni:

	0	1	2
X	2	1	3
Y	3	4	1

Con questi valori di partenza, Mansur può guadagnare la somma maggiore possibile se vende entrambi i suoi cavalli alla fine dell'anno 1 . La sequenza completa di eventi sarà la seguente:

- All'inizio, Mansur ha 1 cavallo.
- Dopo l'anno **0** avrà $1 \cdot X[0] = 2$ cavalli.
- Dopo l'anno **1** avrà $2 \cdot X[1] = 2$ cavalli.
- Si può ora effettuare la vendita. Il guadagno totale sarà $2 \cdot Y[1] = 8$.

Ora, supponiamo ci siano $M = 1$ correzioni: cambiare il valore di $Y[1]$ a **2**. Quindi avremo:

	0	1	2
X	2	1	3
Y	3	2	1

In questo caso, una delle soluzioni ottime è vendere un cavallo dopo l'anno **0** e tre cavalli dopo l'anno **2**. La sequenza completa di eventi sarà la seguente:

- All'inizio, Mansur ha 1 cavallo.
- Dopo l'anno **0** avrà $1 \cdot X[0] = 2$ cavalli.
- Si può ora vendere un cavallo per $Y[0] = 3$, lasciando un cavallo invenduto.
- Dopo l'anno **1** avrà $1 \cdot X[1] = 1$ cavalli.
- Dopo l'anno **2** avrà $1 \cdot X[2] = 3$ cavalli.
- Si possono ora vendere tutti e tre i cavalli per $3 \cdot Y[2] = 3$. Il guadagno totale sarà quindi $3 + 3 = 6$.

Implementazione

Vengono forniti N , X , Y e la lista delle correzioni. Prima della prima correzione, e dopo ogni correzione, dovrai calcolare la quantità massima di denaro che Mansur può guadagnare vendendo i suoi cavalli, modulo $10^9 + 7$. Dovrai implementare le funzioni `init`, `updateX`, and `updateY`.

- `init(N, X, Y)` — Il grader chiamerà questa funzione per prima, ed esattamente una volta.
 - N : il numero di anni.
 - X : un array di lunghezza N , dove $X[i]$ per $0 \leq i \leq N - 1$ è il coefficiente di crescita per l'anno i .
 - Y : un array di lunghezza N , dove $Y[i]$ per $0 \leq i \leq N - 1$ è il prezzo di vendita di un cavallo dopo l'anno i .
 - Nota: sia X che Y specificano i valori iniziali indicati da Mansur (senza applicare alcuna correzione).
 - Dopo che `init` termina, gli array X e Y rimangono validi, e se lo desideri puoi modificarne il contenuto.
 - La funzione deve restituire la massima quantità di denaro che Mansur può guadagnare (modulo $10^9 + 7$) per questi valori iniziali di X e Y .

- `updateX(pos, val)`
 - `pos`: un intero nel range $0, \dots, N - 1$.
 - `val`: il nuovo valore da assegnare a $X[\text{pos}]$.
 - La funzione deve restituire la massima quantità di denaro che Mansur può guadagnare (modulo $10^9 + 7$) dopo aver applicato questa correzione.
- `updateY(pos, val)`
 - `pos`: un intero nel range $0, \dots, N - 1$.
 - `val`: il nuovo valore da assegnare a $Y[\text{pos}]$.
 - La funzione deve restituire la massima quantità di denaro che Mansur può guadagnare (modulo $10^9 + 7$) dopo aver applicato questa correzione.

Puoi assumere che tutti i valori iniziali e corretti di $X[i]$ e $Y[i]$ saranno sempre compresi tra 1 e 10^9 estremi inclusi. Dopo aver chiamato `init`, il grader chiamerà `updateX` e `updateY` svariate volte. Il numero totale delle chiamate a `updateX` e `updateY` sarà M .

Subtask

subtask	punti	N	M	vincoli aggiuntivi
1	17	$1 \leq N \leq 10$	$M = 0$	$X[i], Y[i] \leq 10$, $X[0] \cdot X[1] \cdot \dots \cdot X[N - 1] \leq 1000$
2	17	$1 \leq N \leq 1000$	$0 \leq M \leq 1000$	nessuno
3	20	$1 \leq N \leq 500\,000$	$0 \leq M \leq 100\,000$	$X[i] \geq 2$ e $val \geq 2$, rispettivamente per <code>init</code> e <code>updateX</code>
4	23	$1 \leq N \leq 500\,000$	$0 \leq M \leq 10\,000$	nessuno
5	23	$1 \leq N \leq 500\,000$	$0 \leq M \leq 100\,000$	nessuno

Grader di prova

Il grader di prova legge l'input dal file `horses.in` nel formato seguente:

- riga 1: N
- riga 2: $X[0] \dots X[N - 1]$
- riga 3: $Y[0] \dots Y[N - 1]$
- riga 4: M
- righe 5, ..., $M + 4$: tre interi `type pos val` (`type=1` per `updateX` e `type=2` per `updateY`).

Il grader di prova scrive in output il valore di ritorno di `init`, seguito dal valore di ritorno di tutte le chiamate a `updateX` e `updateY`.