Horses

Mansur ama criar caballos, justamente a el le gusta lo que hicieron sus antepasados. El ahora tiene la mas grande manada en Kazakhstan. Pero este no fue siempre el caso. Hace N años Mansur era justamente un dzhigit (en Kazakh *un hombre joven*) y solamente tenia un caballo simple. El soñaba hacer una gran cantidad de dinero y finalmente convertirse en un bai (en Kazakh *una persona muy rica*).

Tenemos los años desde 0 a $N-1$ en orden cronologico (es decir, el año $N-1$ es el mas reciente). El tiempo de cada año influyó en el crecimiento de la manada. Para cada año i, Mansur recuerda un entero positivo, el coeficiente de crecimiento $X[i]$. Si tu comienzas el año i con h caballos, tu finalizas el año con $h \cdot X[i]$ caballos en la manada.

Los caballos solamete pueden ser vendidos al final de un año. Para cada año i, Mansur recuerda un entero positivo $Y[i]$: el precio para el cual el puede vender un caballo al final de año i. Despues de cada año, fue posible vender arbitrariamente muchos caballos, cada uno en el mismo precio $Y[i]$.

Mansur se pregunta cual es la mayor cantidad de dinero que el pudiera tener ahora si el hubiera elegido el mejor momento para vender sus caballos durante los N años. Usted tiene el honor de ser un invitado de las toi de Mansur (en Kazakh *vacaciones*), y el le pidio a usted responder su pregunta.

La memoria de Mansur mejora a traves de la noche, y asi el hace una secuencia de M actualizaciones. Cada actualizacion cambiará uno de los valores de $X[i]$ o uno de los valores de $Y[i]$. Despues de cada actualizacion el le preguntará cual es la mayor cantidad de dinero que el pudiera haber ganado vendiendo sus caballos. Las actualizaciones de Mansur son acumulativas: cada una de sus respuestas debe tomar en cuenta todas las actualizaciones previas de Mansur. Note que un simple $X[i]$ o $Y[i]$ puede ser actualizado multiples veces.

La respuesta actual a las preguntas de Mansur puede ser muy grande. Para evitar trabajar con numeros grandes, usted solamente requiere reportar las respuestas modulo $10^9 + 7$.

Ejemplo

Suponga que hay $N = 3$ años, con la siguiente información:

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

Para estos valores iniciales, lo que mas puede ganar Mansur es vendiendo sus caballos al final del año 1. El proceso entero será como sigue:

- Inicialmente, Mansur tiene 1 caballo.

- Despues del año 0 el tendrá $1 \cdot X[0] = 2$ caballos.
- Despues del año 1 el tendra $2 \cdot X[1] = 2$ caballos.
- El puede vender esos dos caballos. El provecho total será $2 \cdot Y[1] = 8$.

Entonces, suponga que hay $M = 1$ actualizacion: cambiando $Y[1]$ a 2.

Despues de la actualizacion tendremos:

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

En este caso tendremos, una de las soluciones optimas es vender un caballo despues del año 0 y entonces tres caballos despues del año 2.
El proceso entero será como sigue:

- Inicialmente, Mansur tiene 1 caballo.
- Despues del año 0 el tendrá $1 \cdot X[0] = 2$ caballos.
- El pude vender uno de esos caballos para $Y[0] = 3$, y se quedaria con un caballoleft.
- Despues del año 1 el tendrá $1 \cdot X[1] = 1$ caballos.
- Despues del año 2 el tendra $1 \cdot X[2] = 3$ caballos.
- El puede ahora vender esos tres caballos para $3 \cdot Y[2] = 3$. La cantidad total de dinero es $3 + 3 = 6$.

Tarea

A usted le son dados N, X, Y, y la lista de actualizaciones. Antes de la primera actualizacion, y despues de cada actualizacion, calcule la cantidad maxima de dinero que Mansur puede ganar por sus caballos, modulo $10^9 + 7$. Usted necesita implementar las funciones `init`, `updateX` y `updateY`.

- `init(N, X, Y)` — El grader llamara a esta funcion al inicio y exactamente una vez.
  - `N`: el numero de años.
  - `X`: un arreglo de longitud N. Para $0 \leq i \leq N - 1$, $X[i]$ tiene el coeficiente de crecimiento para cada año.
  - `Y`: un arreglo de longitud N. Para $0 \leq i \leq N - 1$, $Y[i]$ tiene el precio de un caballo despues de cada año.
  - Note que ambos `X` y `Y` especifican los valores iniciales dados por Mansur (antes de cualquier actualizacion).
  - La funcion debe retornar la cantidad maxima de dinero que Mansur puede ganar para estos valores iniciales de X y Y, módulo $10^9 + 7$.
- `updateX(pos, val)`

- `pos`: un entero en el rango de $0, \ldots, N-1$.
- `val`: el nuevo valor para $X[pos]$.
- Las funciones deben retornar la cantidad maxima de dinero que Mansur puede ganar despues de su actualización, módulo $10^9 + 7$.

- `updateY(pos, val)`
  - `pos`: un entero en el rango de $0, \ldots, N-1$.
  - `val`: el nuevo valor para $Y[pos]$.
  - La funcion debe retornar la cantidad maxima de dinero que Mansur puede ganar despues de esta actualizacion, módulo $10^9 + 7$.

Usted puede asumir que todos los valores iniciales, asi como los valores actualizados de $X[i]$ y $Y[i]$ estan entre 1 y 10^9 inclusive.
Despues de llamar `init`, el grader llamara `updateX` y `updateY` varias veces. El numero total de llamadas a `updateX` y `updateY` será M.

Subtareas

subtarea	puntos	N	M	restricciones adicionales
1	17	$1 \le N \le 10$	$M = 0$	$X[i], Y[i] \le 10$, $X[0] \cdot X[1] \cdot \ldots \cdot X[N-1] \le 1,000$
2	17	$1 \le N \le 1,000$	$0 \le M \le 1,000$	ninguno
3	20	$1 \le N \le 500,000$	$0 \le M \le 100,000$	$X[i] \ge 2$ y $val \ge 2$ para `init` y `updateX` correspondientemente
4	23	$1 \le N \le 500,000$	$0 \le M \le 10,000$	ninguno
5	23	$1 \le N \le 500,000$	$0 \le M \le 100,000$	ninguno

Ejemplo de grader

El ejemplo de grader lee la entrada del fichero `horses.in` en el siguiente formato:

- linea 1: `N`
- linea 2: `X[0] ... X[N - 1]`
- linea 3: `Y[0] ... Y[N - 1]`
- linea 4: `M`
- lineas 5, ..., `M` + 4: tres numeros `type pos val` (`type=1` para `updateX` y `type=2` para `updateY`).

El ejemplo de grader imprime el valor de retorno de `init` seguido por los valores de retorno de todas las llamadas a `updateX` y `updateY`.