Caballos

Al igual que todos sus antepasados, Mansur ama la crianza de caballos. En la actualidad es el mayor criador de caballos de Kazajstán. Pero no siempre fue así. Hace N años, Mansur era tan solo un dzhigit (*hombre joven* en Kazajo) y tenía exactamente un caballo. Soñaba con tener mucho dinero y convertirse algún día en bai (*persona muy rica* en Kazajo).

Los años se numeran de 0 a $N-1$ en orden cronológico (es decir, el año $N-1$ es el más reciente). Las condiciones climáticas de cada año afectan la tasa de crecimiento de la cantidad de caballos. Para cada año i, Mansur recuerda un coeficiente de crecimiento entero positivo $X[i]$. Si al comenzar el año i se tienen h caballos, al finalizar se tendrán $h \cdot X[i]$ caballos en total.

Los caballos solo pueden venderse al finalizar el año. Para cada año i, Mansur recuerda un entero positivo $Y[i]$: el precio de venta de un caballo al finalizar el año i. Al terminar cada año, se puede vender una cantidad arbitraria de caballos, todos al mismo precio $Y[i]$.

Mansur se pregunta cuál es la mayor cantidad de dinero que podría tener en la actualidad, si hubiera elegido los mejores momentos para vender sus caballos durante los N años. Tienes el honor de estar invitado al toi de Mansur (*día festivo* en Kazajo), y te ha pedido que respondas esta pregunta.

La memoria de Mansur mejora a lo largo de la fiesta, y por lo tanto realiza una secuencia de M actualizaciones. Cada actualización modificará o bien uno de los valores de $X[i]$, o bien uno de los valores de $Y[i]$. Luego de cada actualización vuelve a preguntarte cuál es la mayor cantidad de dinero que podría haber ganado mediante la venta de caballos. Las actualizaciones de Mansur se acumulan: cada una de tus respuestas debe tener en cuenta todas las actualizaciones previas. Notar que un mismo $X[i]$ o $Y[i]$ podría ser actualizado más de una vez.

Las verdaderas respuestas a las preguntas de Mansur podrían ser enormes. Para evitar trabajar con números muy grandes, solo debes dar el valor de las respuestas módulo 10^9+7.

Ejemplo

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

	0	1	2
X	2	1	3
Y	3	4	1

Para estos valores iniciales, Mansur puede obtener la máxima ganancia si vende ambos caballos al finalizar el año 1. La totalidad del proceso se ve así:

- Inicialmente, Mansur tiene 1 caballo.

- Luego del año 0 tendrá $1 \cdot X[0] = 2$ caballos.
- Luego del año 1 tendrá $2 \cdot X[1] = 2$ caballos.
- Puede vender ahora esos dos caballos. La ganancia total será $2 \cdot Y[1] = 8$.

A continuación, supongamos que hay $M = 1$ actualización: cambiar $Y[1]$ al valor 2.

Luego de la actualización se tiene:

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

En este caso, una solución óptima es vender un caballo al finalizar el año 0 y luego tres caballos al terminar el año 2. La totalidad del proceso se ve así:

- Inicialmente, Mansur tiene 1 caballo.
- Luego del año 0 tendrá $1 \cdot X[0] = 2$ caballos.
- Puede vender ahora uno de esos caballos por $Y[0] = 3$, y quedarse con un caballo.
- Luego del año 1 tendrá $1 \cdot X[1] = 1$ caballo.
- Luego del año 2 tendrá $1 \cdot X[2] = 3$ caballos.
- Puede vender ahora esos tres caballos for $3 \cdot Y[2] = 3$. La ganancia total será $3 + 3 = 6$.

Tarea

Se reciben N, X, Y, y la lista de actualizaciones. Antes de la primera actualización, y luego de cada actualización, se debe computar la máxima cantidad de dinero que Mansur podría conseguir por sus caballos, módulo $10^9 + 7$. Se debe implementar las funciones `init`, `updateX`, y `updateY`.

- `init(N, X, Y)` — El evaluador llamará a esta función exactamente una vez al comienzo.
  - `N`: la cantidad de años.
  - `X`: un arreglo de longitud N. Para $0 \leq i \leq N-1$, $X[i]$ contiene el coeficiente de crecimiento del año i.
  - `Y`: un arreglo de longitud N. Para $0 \leq i \leq N-1$, $Y[i]$ contiene el precio de un caballo luego del año i.
  - Notar que tanto `X` como `Y` especifican los valores iniciales provistos por Mansur (anteriores a todas las actualizaciones).
  - Luego de que `init` termine, los arreglos `X` e `Y` seguirán siendo válidos, y el programa podrá modificar sus contenidos si así lo desea.
  - La función debe retornar la máxima cantidad de dinero que Mansur puede obtener con estos valores iniciales de X e Y, módulo $10^9 + 7$.
- `updateX(pos, val)`

- `pos`: un entero en el rango $0, \ldots, N-1$.
- `val`: el nuevo valor de $X[\text{pos}]$.
- La función debe retornar la máxima cantidad de dinero que Mansur puede obtener luego de esta actualización, módulo $10^9 + 7$.

- `updateY(pos, val)`
  - `pos`: un entero en el rango $0, \ldots, N-1$.
  - `val`: el nuevo valor de $Y[\text{pos}]$.
  - La función debe retornar la máxima cantidad de dinero que Mansur puede obtener luego de esta actualización, módulo $10^9 + 7$.

Se puede asumir que todos los valores (sean iniciales o actualizados) de $X[i]$ e $Y[i]$ están entre 1 y 10^9 inclusive.
Luego de llamar a `init`, el evaluador llamará a `updateX` y `updateY` varias veces. La cantidad total de llamadas a `updateX` y `updateY` será M.

Subtareas

subtarea	puntaje	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	$1 \le N \le 1,000$	$0 \le M \le 1,000$	ninguna
3	20	$1 \le N \le 500,000$	$0 \le M \le 100,000$	$X[i] \ge 2$ y $val \ge 2$ en `init` y `updateX` respectivamente
4	23	$1 \le N \le 500,000$	$0 \le M \le 10,000$	ninguna
5	23	$1 \le N \le 500,000$	$0 \le M \le 100,000$	ninguna

Evaluador de ejemplo

El evaluador de ejemplo lee la entrada del archivo `horses.in` de acuerdo al siguiente formato:

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

El evaluador de ejemplo imprime el valor retornado por la función `init`, seguido por los valores de retorno de las llamadas a `updateX` y `updateY`.