Caballos

A Mansur le fascina la cría de caballos, tal como a sus ancestros. Ahora él tiene la manada más grande en Kazajistán. Pero no siempre fue así. Hace N años, Mansur era apenas un dzhigit (*jovencito*, en Kazajo) y sólo contaba con un caballo. Él soñaba con ganar mucho dinero y finalmente convertirse en un bai (*un hombre muy rico*, en Kazajo).

Enumeremos los años de 0 a $N-1$ en orden cronológico (esto es, el año $N-1$ es el más reciente). El clima de cada año influenciaba el crecimiento de la manada. Por cada año i, Mansur se acuerda de un coeficiente entero positivo de crecimiento, $X[i]$. Si iniciabas el año i con h caballos, terminabas el año con $h \cdot X[i]$ caballos en tu manada.

Los caballos se podían vender exclusivamente al final de un año. Por cada año i, Mansur se acuerda de un entero positivo $Y[i]$: el precio por el cual podía vender un caballo al final del año i. Luego de cada año, era posible vender arbitrariamente muchos caballos, cada uno al mismo precio $Y[i]$.

Mansur se pregunta cuál es la cantidad mayor de dinero que pudiera tener actualmente si ha elegido los mejores momentos para vender sus caballos durantes los N años. Tienes el honor de ser un invitado en el toi (*día feriado*, en Kazajo) de Mansur, y te ha pedido que respondas esta pregunta.

La memoria de Mansur se mejora conforme pasa la noche, así que él efectúa una serie de M actualizaciones. Cada actualización cambiaría uno de los valores $X[i]$ o uno de los valores $Y[i]$. Luego de cada actualización, él te pregunta de nuevo cuál es la mayor cantidad de dinero que pudiera tener por la venta de sus caballos. Las actualizaciones de Mansur son acumulativas: cada una de tus respuestas debería tomar en cuenta todas las actualizaciones previas. Toma en cuenta que cualquier $X[i]$ o $Y[i]$ puede ser actualizado más de una vez.

Las respuestas de Mansur pueden ser gigantescas. Para evitar trabajar con números grandes, se te requiere reportar las respuestas módulo $10^9 + 7$.

Ejemplo

Supón 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 cantidad de dinero si vende sus dos caballos al final del año 1. El proceso completo se detalla a continuación:

- Inicialmente, Mansur tiene 1 caballo.

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

Luego, supón que hay $M = 1$ actualización: cambiar $Y[1]$ a 2.

Luego de la actualización tendremos:

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

En este caso, una de las soluciones óptimas es vender un caballo luego del año 0 y luego tres caballos luego del año 2.
El proceso completo se detalla de la siguiente manera:

- Inicialmente, Mansur tiene 1 caballo.
- Luego del año 0 él tendrá $1 \cdot X[0] = 2$ caballos.
- Ahora puede vender uno de esos caballos por $Y[0] = 3$, y se quedaría con uno.
- Luego del año 1 tendrá $1 \cdot X[1] = 1$ caballo.
- Luego del año 2 tendrá $1 \cdot X[2] = 3$ caballos.
- Ahora puede vender esos tres caballos por $3 \cdot Y[2] = 3$. La cantidad de dinero total sería $3 + 3 = 6$.

Tarea

Se te proveerá N, X, Y, y la lista de actualizaciones. Antes de la primera de estas, y luego de cada actualización, calcula la máxima cantidad de dinero que Mansur puede obtener por sus caballos, módulo $10^9 + 7$. Necesitas implementar ls funciones `init`, `updateX` y `updateY`.

- `init(N, X, Y)` — El grader llamará a esta función primero y sólo una vez.
  - `N`: el número 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 luego de cada año.
  - Toma nota que tanto `X` como `Y` especifican los valores iniciales dados por Mansur (antes de cualquier actualización).
  - La función debe retornar la máxima cantidad de dinero que Mansur pudiera obtener dados estos valores iniciales de X y Y, módulo $10^9 + 7$.
- `updateX(pos, val)`

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

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

Puedes asumir que todos los valores iniciales y actualizados para $X[i]$ y $Y[i]$ estarán entre 1 y 10^9 inclusive.
Luego de llamar a `init`, el grader llamará a `updateX` y `updateY` varias veces. El número total de llamadas a `updateX` y `updateY` será M.

Sub-tareas

sub-tarea	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	$1 \le N \le 1,000$	$0 \le M \le 1,000$	ninguna
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` 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

Grader de Ejemplo

El grader de ejemplo leerá la entrada del archivo `horses.in`, la cual estará en el 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 grader de ejemplo imprimirá el valor de retorno de `init` seguido por los valores de retorno de todas las llamadas a `updateX` y `updateY`.