Caballos

Mansur adora criar caballos tal como sus antepasados lo hicieron. Ahora él tiene la manada más grande de caballos de Kazajistán. Sin embargo no siempre fue así. N años atrás, Mansur era sólo un dzhigit (*Hombre joven* en Kazajo) y sólo tenía un caballo. Su sueño era generar mucho dinero y finalmente volverse un bai (*hombre rico* en Kazajo).

Vamos a numerar los años de 0 a $N-1$ en orden cronológico (es decir, el año $N-1$ es el año más reciente). El crecimiento de la manada está influenciado por el clima de cada año. Para cada año i, Mansur recuerda un coeficiente entero positivo $X[i]$. Es decir, si inició en el año i con h caballos, al final del año tendrá $h \cdot X[i]$ caballos en la manada.

Los caballos sólo pueden ser vendidos al final del año. Para cada año i, Mansur recuerda un entero positivo $Y[i]$: el precio en el cual un caballo puede ser vendido al final del año i. Después de cada año, es posible vender cualquier cantidad de caballo, cada uno al precio $Y[i]$.

Mansur se pregunta cuál es la máxima cantidad de dinero que pudo haber obtenido si hubiera escogido los mejores momentos para vender sus caballos durante los últimos N años. Tienes el honor de ser invitado en el tou de Mansur (*cumpleaños* de Mansur, en Kazajo) y él te ha pedido si le puedes contestar esa pregunta.

La memoria de Mansur ha estado mejorando durante el día, y te ha dado una secuencia M de actualizaciones. Cada actualización cambiará algún valor de $X[i]$ o algún valor de $Y[i]$. Después de cada actualización, Mansur te preguntará lo descrito anteriormente: cuál es la máxima cantidad de dinero que pudo haber obtenido vendiendo correctamente sus caballos. Las actualizaciones de Mansur se van acumulando, es decir para cada pregunta que respondas, tienes que tomar en cuenta todas las actualizaciones que se hicieron anteriormente. Es importante notar que más de una actualización puede modificar el mismo $X[i]$ o $Y[i]$ varias veces.

Las respuestas a las preguntas de Mansur pueden ser muy grandes. Para evitar trabajar con números grandes, tienes que reportar la respuesta modulo $10^9 + 7$.

Ejemplo

Supón que inicialmente 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 ganar la mayor cantidad de dinero si vende sus dos caballos al final del año 1. El proceso entero se vería de la siguiente manera.

- Al inicio Mansur tiene un caballo.
- Al final del año 0, él tendrá $1 \cdot X[0] = 2$ caballos.
- Al final del año 1, él tendrá $1 \cdot X[1] = 2$ caballos.
- En este momento, puede vender sus dos caballos. La ganancia total será $2 \cdot Y[1] = 8$.

Ahora supongamos que llega una actualización $M = 1$: el valor de $Y[1]$ cambia a $Y[1] = 2$.

Es decir, después de la actualización tendremos:

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

Es este caso, la solución óptima es vender un caballo al final del año 0 y vender los otros 3 caballo al final del año 2.

El proceso entero se vería de la siguiente manera.

- Al inicio Mansur tiene un caballo.
- Al final del año 0 el tendrá $1 \cdot X[0] = 2$ caballos.
- En este momento, puede vender uno de sus caballos por $Y[0] = 3$ y le quedaría un caballo.
- Al final del año 1 el tendrá $1 \cdot X[1] = 1$ caballos.
- Al final del año 2 el tendrá $1 \cdot X[2] = 3$ caballos.
- Ahora puede vender los 3 caballos por $3 \cdot Y[2] = 3$. La ganancia total sería $3 + 3 = 6$.

Tarea

Se te da N, X, Y y la lista de actualizaciones. Antes de la primera actualización y después de cada actualización, calcula la máxima cantidad de dinero que Mansur puede obtener vendiendo sus caballos, modulo $10^9 + 7$.

Tienes que implementar las funciones `init`, `updateX` y `updateY`.

- `init(N, X, Y)` — El evaluador llamará esta función exactamente una vez.
  - `N`: el número de años.
  - `X`: un arreglo de longitud N, donde $0 \leq i \leq N - 1$, $X[i]$ denota el coeficiente de crecimiento de cada año i.
  - `Y`: un arreglo de longitud N, donde $0 \leq i \leq N - 1$, $Y[i]$ denota el costo de cada caballo al final de cada año i.
  - Es importante notar que `X` y `Y`especifican los valores iniciales de Mansur (antes de cualquier actualización).
  - Después de que `init` termina, los arreglos `X` y `Y` siguen siendo validos y puedes modificar

sus contenidos como gustes.

- La función debe regresar la máxima cantidad de dinero que Mansur pudo haber obtenido para los valores iniciales de X y Y, modulo $10^9 + 7$.

- `updateX(pos, val)`
  - `pos`: un entero en el rango $0, \ldots, N-1$.
  - `val`: el nuevo valor para $X[\texttt{pos}]$.
  - La función debe regresar la máxima cantidad de dinero que Mansur pudo haber obtenido después de esta actualización, modulo $10^9 + 7$.
- `updateY(pos, val)`
  - `pos`: un entero en el rango $0, \ldots, N-1$.
  - `val`: el nuevo valor para $Y[\texttt{pos}]$.
  - La función debe regresar la máxima cantidad de dinero que Mansur pudo haber obtenido después de esta actualización, modulo $10^9 + 7$.

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

Subtarea

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	$1 \le N \le 1,000$	$0 \le M \le 1,000$	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

Evaluador de ejemplo

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

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

`updateY`).

El evaluador de ejemplo imprime el valor que regresa `init` seguido de los valores que regresan las llamadas a `updateX` y `updateY`.