Pferde

Mansur liebt es, Pferde zu züchten, genauso wie es seine Vorfahren schon immer taten. Er hat nun die grösste Herde in Kasachstan. Aber das war nicht immer so. Vor N Jahren war Mansur noch ein Dzhigit (kasachisch für *junger Mann*) und besass nur ein einziges Pferd. Er träumte davon, viel Geld zu verdienen und ein Bai (kasachisch für *sehr reiche Person*) zu werden.

Wir nummerieren die Jahre in chronologischer Reihenfolge von 0 bis $N-1$ (d.h. das Jahr $N-1$ ist das letzte). In jedem Jahr hat das Wetter das Wachstum der Herde beeinflusst. Für jedes Jahr i erinnert sich Mansur an einen positiven, ganzzahligen Wachstumskoeffizienten $X[i]$. Wer am Anfang des Jahres i genau h Pferde hatte, dessen Herde hatte am Ende des Jahres genau $h \cdot X[i]$ Pferde.

Pferde können nur am Ende eines Jahres verkauft werden. Mansur erinnert sich für jedes Jahr i an eine positive Ganzzahl $Y[i]$: den Verkaufspreis pro Pferd am Ende des Jahres i. Nach jedem Jahr war es möglich, beliebig viele Pferde zu verkaufen, jedes Pferd zum selben Preis $Y[i]$.

Mansur möchte gerne wissen, was der grösste Geldbetrag ist, den er jetzt haben könnte, falls er während der N Jahre die besten Momente zum Verkauf seiner Pferde gewählt hätte. Du hast die Ehre, ein Gast auf Mansurs Toi (kasachisch für *Feier*) zu sein, und er hat dich aufgefordert, diese Frage zu beantworten.

Mansurs Erinnerungen verbessern sich im Laufe des Abends und deshalb macht er eine Folge von M Updates. Jedes Update verändert entweder einen der Werte $X[i]$ oder $Y[i]$. Nach jedem Update fragt er dich wieder nach dem grössten Geldbetrag, den er durch den Verkauf seiner Pferde hätte verdienen können. Mansurs Updates sind kumulativ: Jede deiner Antworten soll alle bisherigen Updates berücksichtigen. Beachte, dass ein einzelnes $X[i]$ oder $Y[i]$ mehrmals verändert werden kann.

Die tatsächlichen Antworten auf Mansurs Fragen können riesig sein. Um den Umgang mit grossen Zahlen zu vermeiden, musst du die Antwort nur modulo $10^9 + 7$ ausgeben.

Beispiel

Nehmen wir an, dass es $N = 3$ Jahre gibt und wir folgende Informationen haben:

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

Für diese Anfangswerte kann Mansur am meisten verdienen, wenn er seine beiden Pferde am Ende des Jahres 1 verkauft. Der ganze Ablauf sieht so aus:

- Zu Beginn besitzt Mansur 1 Pferd.

- Am Ende von Jahr 0 besitzt er $1 \cdot X[0] = 2$ Pferde.
- Am Ende von Jahr 1 besitzt er $2 \cdot X[1] = 2$ Pferde.
- Er kann diese beiden Pferde nun verkaufen. Der gesamte Verkaufspreis beträgt $2 \cdot Y[1] = 8$.

Nehmen wir an, dass es danach genau ein Update gibt ($M = 1$). Dieses Update ändert $Y[1]$ zu 2.

Nach dem Update haben wir:

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

In diesem Fall besteht eine der optimalen Lösungen darin, ein Pferd nach Jahr 0 und drei Pferde nach Jahr 2 zu verkaufen. Der ganze Ablauf sieht so aus:

- Zuerst hat Mansur 1 Pferd.
- Am Ende von Jahr 0 besitzt er $1 \cdot X[0] = 2$ Pferde.
- Er verkauft nun eines dieser Pferde für $Y[0] = 3$ und hat ein Pferd übrig.
- Am Ende von Jahr 1 besitzt er $1 \cdot X[1] = 1$ Pferd.
- Am Ende von Jahr 2 besitzt er $1 \cdot X[2] = 3$ Pferde.
- Er kann nun diese drei Pferde für $3 \cdot Y[2] = 3$ verkaufen. Der gesamte Verkaufspreis beträgt $3 + 3 = 6$.

Aufgabe

Du bekommst N, X, Y und die Liste der Updates. Vor dem ersten Update und nach jedem Update sollst du den maximalen Geldbetrag bestimmen, welchen Mansur für seine Pferde bekommen kann, modulo $10^9 + 7$. Du musst die Funktionen `init`, `updateX`, und `updateY` implementieren.

- `init(N, X, Y)` — Der Grader ruft diese Funktion als Erstes und genau einmal auf.
  - `N`: die Anzahl Jahre.
  - `X`: ein Array der Länge N. Für $0 \leq i \leq N - 1$ bezeichnet $X[i]$ den Wachstumskoeffizienten im Jahr i.
  - `Y`: ein Array der Länge N. Für $0 \leq i \leq N - 1$ bezeichnet $Y[i]$ den Preis pro Pferd am Ende von Jahr i.
  - Beachte, dass `X` und `Y` die Anfangswerte beschreiben, welche du von Mansur bekommst (vor allen Updates).
  - Nachdem `init` terminiert, bleiben die Arrays `X` und `Y` gültig und du darfst deren Inhalt verändern, falls du möchtest.
  - Die Funktion soll den maximalen Geldbetrag zurückgeben, welchen Mansur mit diesen Anfangswerten für X und Y bekommen kann, modulo $10^9 + 7$.

- `updateX(pos, val)`
  - `pos`: eine Ganzzahl aus dem Bereicht $0, \ldots, N-1$.
  - `val`: der neue Wert für $X[\text{pos}]$.
  - Die Funktion soll den maximalen Geldbetrag zurückgeben, welchen Mansur nach diesem Update bekommen kann, modulo $10^9 + 7$.
- `updateY(pos, val)`
  - `pos`: eine Ganzzahl aus dem Bereicht $0, \ldots, N-1$.
  - `val`: der neue Wert für $Y[\text{pos}]$.
  - Die Funktion soll den maximalen Geldbetrag zurückgeben, welchen Mansur nach diesem Update bekommen kann, modulo $10^9 + 7$.

Du darfst annehmen, dass alle anfänglichen wie auch alle veränderten Werte von $X[i]$ und $Y[i]$ zwischen 1 und 10^9 (inklusive) liegen.
Nach dem Aufruf von `init` wird der Grader `updateX` und `updateY` mehrmals aufrufen. `updateX` und `updateY` werden zusammengezählt M Mal aufgerufen.

Teilaufgaben

Teilaufgabe	Punkte	N	M	Weitere Beschränkungen
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	$1 \le N \le 1{,}000$	$0 \le M \le 1{,}000$	keine
3	20	$1 \le N \le 500{,}000$	$0 \le M \le 100{,}000$	$X[i] \ge 2$ und `val` ≥ 2 für `init` und `updateX` respektive
4	23	$1 \le N \le 500{,}000$	$0 \le M \le 10{,}000$	keine
5	23	$1 \le N \le 500{,}000$	$0 \le M \le 100{,}000$	keine

Beispiel-Grader

Der Beispiel-Grader liest die Eingabe aus der Datei `horses.in` in dem folgenden Format:

- Zeile 1: `N`
- Zeile 2: `X[0]` … `X[N - 1]`
- Zeile 3: `Y[0]` … `Y[N - 1]`
- Zeile 4: `M`
- Zeilen 5, …, `M` + 4: drei Zahlen `type pos val` (`type=1` für `updateX` und `type=2` für `updateY`).

Der Beispiel-Grader gibt den Return-Wert von `init` aus, gefolgt von den Return-Werten aller Aufrufe von `updateX` und `updateY`.