



말

만수르는 조상들과 마찬가지로 말을 키우는 것을 좋아한다. 그는 카자흐스탄에서 말을 제일 많이 갖고 있다. 그렇지만 꼭 항상 그랬던 것은 아닌데, N 년 전에만 해도 만수르는 그저 젊은이일 뿐이어서 말이 한마리밖에 없었다. 만수르는 돈을 많이 벌어서 부자가 되고 싶었다.

시간순으로 매 해를 0번 해부터 $N - 1$ 번 해로 번호를 매기자. (즉, $N - 1$ 번 해가 가장 최근이다.) 해마다 낳는 말이 자라는데 영향을 미친다. 만수르가 기억하기로는, i 번 해에 말의 마릿수가 늘어난 비율은 양의 정수 $X[i]$ 이다. 만약 i 번 해 연초에 말이 h 마리가 있었다면, 그 해 연말에는 말이 모두 $h \cdot X[i]$ 마리가 된다.

말은 매해 연말에만 팔 수 있다. 만수르가 기억하기로 i 번 해의 말값은 양의 정수 $Y[i]$ 였다. 즉, 매해 연말에 갖고 있는 말 중 팔 수 있는 말의 수에는 제약이 없고, 말 한마리 값은 $Y[i]$ 로 모두 같다.

만수르는 지난 N 년 동안, 말을 파는 시기를 잘 정했다면 얼마나 많은 돈을 벌 수 있었을지가 궁금해졌다. 당신이 만수르를 방문했을 때 이 질문을 받게 되었다.

저녁동안 만수르의 기억은 점점 정확해져서, 총 M 번의 수정을 하게 된다. 수정을 한 번 할 때마다 $X[i]$ 의 값 중 하나, 또는 $Y[i]$ 의 값 중 하나가 바뀐다. 수정을 한 번 할 때마다 만수르는 말을 팔아서 벌 수 있는 돈의 최대값을 물어본다. 만수르가 수정할 때마다, 수정된 내용들은 누적된다. 즉, 만수르에게 대답할 때는 지금까지 만수르가 한 수정들을 모두 반영해야 한다. 한 $X[i]$ 또는 $Y[i]$ 가 여러 번 수정되는 경우도 가능하다.

만수르의 질문에 대한 답은 매우 큰 수일 수 있다. 큰 수를 다룰 때 생기는 문제를 피하기 위해서, 답을 $10^9 + 7$ 로 나눈 나머지를 알려주면 된다.

예제

$N = 3$ 년에 대해서 다음과 같은 정보가 주어졌다고 하자.

	0	1	2
X	2	1	3
Y	3	4	1

이 초기 정보를 가지고, 만수르는 1번 해 연말에 모든 말을 다 팔았다면 가장 많은 돈을 벌 수 있다. 전체 과정은 다음과 같다.

- 처음에 만수르는 말이 한 마리 있다.
- 0번 해 연말에 만수르는 $1 \cdot X[0] = 2$ 마리의 말이 있다.
- 1번 해 연말에 만수르는 $2 \cdot X[1] = 2$ 마리의 말이 있다.
- 이제 말 두 마리를 모두 팔 수 있다. 전체 이익은 $2 \cdot Y[1] = 8$ 이다.

이제, $M = 1$ 번 수정을 하여 $Y[1]$ 이 2로 바뀌었다.
수정 후의 정보는 다음과 같다.

	0	1	2
X	2	1	3
Y	3	2	1

이 경우, 최적해 중 하나는 한 마리를 0번 해 연말에 팔고 2번 해 연말에 세 마리를 파는 것이다. 전체 과정은 다음과 같다

- 처음에 만수르는 말이 한 마리 있다.
- 0번 해 연말에 만수르는 $1 \cdot X[0] = 2$ 마리의 말이 있다.
- 이제 이 중 한 마리를 $Y[0] = 3$ 에 팔 수 있고, 한 마리가 남아 있다.
- 1번 해 연말에 만수르는 $1 \cdot X[1] = 1$ 마리의 말이 있다.
- 2번 해 연말에 만수르는 $1 \cdot X[2] = 3$ 마리의 말이 있다.
- 이제 말 세 마리를 모두 $3 \cdot Y[2] = 3$ 에 팔 수 있다. 전체 이익은 $3 + 3 = 6$ 이다.

문제

N, X, Y 와 수정된 내용의 리스트가 주어진다. 첫번째 수정을 하기 전과, 매번 수정을 한 다음에 대해서 만수르가 말을 팔아서 벌 수 있는 돈의 최대값을 $10^9 + 7$ 로 나눈 나머지를 구하라. 이를 위해서 함수 `init`, `updateX`, `updateY`를 구현해야 한다.

- `init(N, X, Y)` — 그레이더는 이 함수를 맨 처음 정확히 한 번 호출한다.
 - `N`: 전체 해 수
 - `x`: 길이 N 인 배열. $0 \leq i \leq N - 1$ 일 때, $X[i]$ 는 i 번 해 연말에 말의 마릿수가 늘어난 비율이다.
 - `y`: 길이 N 인 배열. $0 \leq i \leq N - 1$ 일 때, $Y[i]$ 는 i 번 해 연말에 말 한마리의 값이다.
 - `x`와 `y`는 만수르가 처음에 준 값을 나타낸다. (수정을 하기 전의 값)
 - `init` 함수가 종료한 후, 배열 `x`와 `y`는 유효한 주소에 있으며, 원한다면 이 배열의 내용을 수정할 수 있다.
 - 이 함수는 X 와 Y 의 초기값을 가지고 만수르가 말을 팔아서 벌 수 있는 돈의 최대값을 $10^9 + 7$ 로 나눈 나머지를 리턴해야 한다.
- `updateX(pos, val)`
 - `pos`: $0, \dots, N - 1$ 범위 내의 정수.
 - `val`: $X[pos]$ 의 새로운 값.
 - 이 함수는 주어진 정보에 따라 수정을 한 후, 만수르가 말을 팔아서 벌 수 있는 돈의 최대값을 $10^9 + 7$ 로 나눈 나머지를 리턴해야 한다.

- `updateY(pos, val)`
 - `pos`: $0, \dots, N - 1$ 범위 내의 정수.
 - `val`: $Y[pos]$ 의 새로운 값.
 - 이 함수는 주어진 정보에 따라 수정을 한 후, 만수르가 말을 팔아서 벌 수 있는 돈의 최대값을 $10^9 + 7$ 로 나눈 나머지를 리턴해야 한다.

모든 $X[i]$ 와 $Y[i]$ 의 값은 항상 (즉, 처음에도, 매번 수정을 한 다음에도) 1 이상 10^9 이하를 만족한다.

`init`을 호출한 후, 그레이더는 `updateX`와 `updateY`를 여러 번 호출할 것이다. `updateX`와 `updateY`의 전체 호출 횟수는 M 이다.

부분문제

부분문제	점수	N	M	추가적인 제약조건
1	17	$1 \leq N \leq 10$	$M = 0$	$X[i], Y[i] \leq 10$, $X[0] \cdot X[1] \cdot \dots \cdot X[N - 1] \leq 1,000$
2	17	$1 \leq N \leq 1,000$	$0 \leq M \leq 1,000$	없음
3	20	$1 \leq N \leq 500,000$	$0 \leq M \leq 100,000$	<code>init</code> 과 <code>updateX</code> 에 대해서 각각 $X[i] \geq 2$ 이고 $val \geq 2$
4	23	$1 \leq N \leq 500,000$	$0 \leq M \leq 10,000$	없음
5	23	$1 \leq N \leq 500,000$	$0 \leq M \leq 100,000$	없음

Sample grader

Sample grader는 다음 양식에 따라 `horses.in` 파일에서 입력을 읽어들이는다.

- line 1: N
- line 2: $X[0] \dots X[N - 1]$
- line 3: $Y[0] \dots Y[N - 1]$
- line 4: M
- lines 5, ..., $M + 4$: `type pos val`에 해당하는 세 숫자 (`type=1`이면 `updateX`이고 `type=2`이면 `updateY`).

Sample grader는 처음 `init`의 리턴값을 출력하고, 차례로 `updateX`와 `updateY`의 모든 호출에 대한 리턴값을 출력한다.