Problem Buses from Bragadiru

Input file	`stdin`
Output file	`stdout`

Bragadiru, a city on the outskirts of Bucharest, has just revamped its bus network! Before leaving for Piatra Neamț, Andrei has to take the bus from his station to another stop, where he will meet another Andrei (from the legendary team *Andrei* of 2024). He observed that the city hall has imposed a new pricing model for the buses:

- Each stop i has two associated costs: $depart_i$, the price to leave from station i, and $arrive_i$, the price to arrive at station i.
- Traveling directly from station i to station j costs $C(i,j) = depart_i \cdot arrive_j$ lei.
- A bus trip is a sequence of stops $X_0, X_1, \ldots, X_k$ where each consecutive pair is connected by a direct hop **(stops may repeat)**. Its **length** is k – the number of hops – and its total cost is the product $C(X_0, X_1) \cdot C(X_1, X_2) \cdots C(X_{k-1}, X_k)$ lei.

Task

Andrei watches a stream of Q events. Some are pricing adjustments published by the city hall (the prices change in real time); others are questions Andrei asks about the network in its current state. Each event has one of the following forms:

- `1 L R val` – *(pricing update)* $depart_i \leftarrow depart_i + val$ for all $L \le i \le R$.
- `2 L R val` – *(pricing update)* $arrive_i \leftarrow arrive_i + val$ for all $L \le i \le R$.
- `3 L R k` – *(query)* Restrict the network to only the stops in the interval $[L, R]$. Compute the sum of total costs of **all** trips of length exactly k whose stops lie entirely within $[L, R]$.

Since the answer to a query can be very large, output it modulo $10^9 + 7$.

Input

The first line of the input contains an integer N, representing the number of bus stops in Bragadiru.

The second line contains N space-separated integers, $depart_1, depart_2, \ldots, depart_N$, representing the cost to depart from each stop.

The third line contains N space-separated integers, $arrive_1, arrive_2, \ldots, arrive_N$, representing the cost to arrive at each stop.

The fourth line contains an integer Q, representing the number of events.

The following Q lines describe the events. Each line begins with an integer $type \in \{1, 2, 3\}$, followed by additional space-separated integers corresponding to the parameters described above.

Output

For each query event (type 3), output on a separate line the requested sum, taken modulo $10^9 + 7$.

Restrictions

- $1 \le N, Q \le 100\,000$
- $1 \le arrive_i, depart_i \le 10\,000$ for all $1 \le i \le N$.
- $1 \le L \le R \le N$ for every event.
- $1 \le val \le 10\,000$ for every event of type 1 or 2.
- $1 \le k \le 1\,000\,000\,000$ for every event of type 3.

#	Points	Restrictions
1	0	Examples
2	10	$N, Q \leq 1000$ and $k \leq 20$ for every query.
3	13	$N, Q \leq 100$.
4	15	$N, Q \leq 5000$ and $k \leq 100$ for every query.
5	7	$k = 1$ for every query.
6	15	No update events occur – only queries.
7	17	Only one of the two update types (type 1 or type 2) appears in the input.
8	23	No further restrictions.

Examples

Input file	Output file
3 2 3 1 1 2 4 4 3 1 3 1 1 2 2 5 3 1 3 1 3 2 3 2	42 77 1080

Explanation

Initially $depart = [2, 3, 1]$ and $arrive = [1, 2, 4]$.

- `3 1 3 1`: query on $[1, 3]$ with $k = 1$. Each trip has a single hop $i \to j$ for $i, j \in \{1, 2, 3\}$, with cost $depart_i \cdot arrive_j$. Summing over all 9 pairs gives 42.
- `1 2 2 5`: update – $depart_2$ becomes $3 + 5 = 8$, so $depart = [2, 8, 1]$.
- `3 1 3 1`: same kind of query, now $(2 + 8 + 1) \cdot (1 + 2 + 4) = 11 \cdot 7 = 77$.
- `3 2 3 2`: query on $[2, 3]$ with $k = 2$. A trip of length 2 visits a sequence of **three** stops $X_0 \to X_1 \to X_2$, each in $\{2, 3\}$, and **stops may repeat** – e.g. $2 \to 2 \to 2$ or $3 \to 3 \to 2$ are perfectly valid trips. So there are $2^3 = 8$ trips in total, not just the ones that use distinct stops.

With $depart = [2, 8, 1]$ and $arrive = [1, 2, 4]$, the four edge costs within $\{2, 3\}$ are

$$C(2,2) = 8 \cdot 2 = 16, \quad C(2,3) = 8 \cdot 4 = 32, \quad C(3,2) = 1 \cdot 2 = 2, \quad C(3,3) = 1 \cdot 4 = 4.$$

Each trip's cost is the product of its two edges:

$$\begin{array}{llll} 2 \to 2 \to 2 : 16 \cdot 16 = 256 & 2 \to 2 \to 3 : 16 \cdot 32 = 512 & 2 \to 3 \to 2 : 32 \cdot 2 = 64 & 2 \to 3 \to 3 : 32 \cdot 4 = 128 \\ 3 \to 2 \to 2 : 2 \cdot 16 = 32 & 3 \to 2 \to 3 : 2 \cdot 32 = 64 & 3 \to 3 \to 2 : 4 \cdot 2 = 8 & 3 \to 3 \to 3 : 4 \cdot 4 = 16 \end{array}$$

Their sum is $256 + 512 + 64 + 128 + 32 + 64 + 8 + 16 = 1080$.