# Zelda

| | |
|---|---|
| Input file: | `standard input` |
| Output file: | `standard output` |
| Time limit: | 3 seconds |
| Memory limit: | 512 megabytes |

Zelda is given integers $l, r$ $(l \le r)$ and she has two sequences $a$ and $b$ of length $n$, indexed from 1 to $n$. Zelda will use these two sequences to play a game, which proceeds as follows:

- Uniformly and randomly select an integer $x$ from the interval $[l, r]$.
- For $i = 1, 2, \cdots, n$, execute the following operations in order: first let $x \leftarrow \min(x, a_i)$, then let $x \leftarrow \max(x, b_i)$.

Zelda wants to know the expected value of the final $x$.

Unfortunately, Zelda has forgotten some of the integers in sequences $a$ and $b$, so she wants you to calculate the expected value of the final $x$ when all forgotten integers are also uniformly and randomly selected from the interval $[l, r]$, modulo $10^9 + 7$. Since Zelda will play the game multiple times, you need to answer queries for multiple different values of $r$.

## Input

The first line of the input contains three integers $n, q, l$ $(1 \le n, l \le 200,\ 1 \le q \le 5 \times 10^4)$, indicating the length of the two sequences, the number of times Zelda plays the game, and the lower bound for uniform random selections.

The next line of the input contains $n$ non-negative integers $a_1, a_2, \cdots, a_n$ ($a_i = 0$ or $l \le a_i \le 200$). $a_i = 0$ means the integer was forgotten by Zelda.

The next line of the input contains $n$ non-negative integers $b_1, b_2, \cdots, b_n$ ($b_i = 0$ or $l \le b_i \le 200$). $b_i = 0$ means the integer was forgotten by Zelda.

The next line of the input contains $q$ integers $r_1, r_2, \cdots, r_q$ $(l \le r_i \le 10^9)$, indicating the value of $r$ each time Zelda plays a game.

## Output

The output contains $q$ lines, where the $i$-th line contains a single integer representing the answer for the $i$-th game, modulo $10^9 + 7$.

Formally, let $M = 10^9 + 7$. It can be shown that the answer can be expressed as an irreducible fraction $\frac{p}{q}$, where $p$ and $q$ are integers and $q \not\equiv 0 \pmod{M}$. Output the integer equal to $p \cdot q^{-1} \pmod{M}$. In other words, output such an integer $x$ that $0 \le x < M$ and $x \cdot q \equiv p \pmod{M}$.

## Examples

| standard input | standard output |
|---|---|
| 2 5 1<br>2 2<br>0 1<br>1 2 3 50 538 | 1<br>750000007<br>888888897<br>857200008<br>228862927 |
| 3 5 2<br>3 3 0<br>3 0 0<br>2 3 4 50 519 | 2<br>750000008<br>962962973<br>798646858<br>311741862 |

## Note

In the first example, sequences $a = [2, 2]$ and $b = [0, 1]$ (where $b_1 = 0$ indicates a forgotten value), and there are $q = 5$ queries.

In the first query, $l = r = 1$, so $x = b_1 = 1$ always. The game process is as follows:

- Initially, $x = 1$.
- Do operations with $i = 1$, $x \leftarrow \min(x, a_1) = 1$ first, and then $x \leftarrow \max(x, b_1) = 1$.
- Do operations with $i = 2$, $x \leftarrow \min(x, a_2) = 1$ first, and then $x \leftarrow \max(x, b_2) = 1$.

So the answer for the first query is 1.

In the second query, $l = 1, r = 2$, so $x, b_1$ should be selected from $[1, 2]$.

- $x = 1, b_1 = 1$, $x$ becomes 1 after all operations.
- $x = 1, b_1 = 2$, $x$ becomes 2 after all operations.
- $x = 2, b_1 = 1$, $x$ becomes 2 after all operations.
- $x = 2, b_1 = 2$, $x$ becomes 2 after all operations.

Each case has probability $\frac{1}{4}$, so the answer for the second query is $\frac{1}{4} \times (1 + 2 + 2 + 2) = \frac{7}{4}$.