Little Walter likes playing with his toy scales. He has N types of weights. The i^{th} weight type has weight a_i. There are infinitely many weights of each type.

Recently, Walter defined a function, $F(X)$, denoting the number of different ways to combine several weights so their total weight is equal to X. Ways are considered to be different if there is a type which has a different number of weights used in these two ways.

For example, if there are 3 types of weights with corresonding weights 1, 1, and 2, then there are 4 ways to get a total weight of 2:

1. Use 2 weights of type 1.
2. Use 2 weights of type 2.
3. Use 1 weight of type 1 and 1 weight of type 2.
4. Use 1 weight of type 3.

Given N, L, R, and $a_1, a_2, \ldots, a_N$, can you find the value of $F(L) + F(L+1) + \ldots + F(R)$?

Input Format

The first line contains a single integer, N, denoting the number of types of weights.
The second line contains N space-separated integers describing the values of $a_1, a_2, \ldots, a_N$, respectively
The third line contains two space-separated integers denoting the respective values of L and R.

Constraints

- $1 \le N \le 10$
- $0 < a_i \le 10^5$
- $a_1 \times a_2 \times \ldots \times a_N \le 10^5$
- $1 \le L \le R \le 10^{17}$

Note: The time limit for C/C++ is 1 second, and for Java it's 2 seconds.

Output Format

Print a single integer denoting the answer to the question. As this value can be very large, your answer must be modulo $10^9 + 7$.

Sample Input

```
3
1 2 3
1 6
```

Sample Output

```
22
```

Explanation

$F(1) = 1$
$F(2) = 2$
$F(3) = 3$
$F(4) = 4$
$F(5) = 5$
$F(6) = 7$