Problem I – Isaac and MOD Convolution

Isaac is a passionate competitive programmer, especially fascinated by convolution techniques used in ICPC problems. During the lectures of last year's *Training Camp Mexico* (TCMX for short), he mastered *bitwise* convolutions:

- **AND convolution**: for each k,

$$C[k] = \sum_{i \,\&\, j = k} A[i] \cdot B[j]$$

- **OR convolution**: for each k,

$$C[k] = \sum_{i \,|\, j = k} A[i] \cdot B[j]$$

- **XOR convolution**: for each k,

$$C[k] = \sum_{i \oplus j = k} A[i] \cdot B[j]$$

He implemented all of them successfully with SOS DP and used them to solve problems involving fast subset transforms.

This year, during the *Tercera Fecha del Gran Premio de México ICPC 2025*, which coincides with the third contest of this year's TCMX, Isaac is facing a problem involving a new and intriguing type: MOD convolution. It is defined similarly to the bitwise convolutions, but replacing the bitwise index condition with a modular remainder condition:

$$C[k] = \sum_{\substack{1 \le i \le n \\ k < j \le m \\ i \equiv k \bmod j}} A[i] \cdot B[j]$$

As everyone knows, MOD convolution can be easily calculated using [redacted].

The problem asks, given arrays A of length n and B of length m, to compute:

$$\sum_{k=0}^{\infty} k \cdot C[k].$$

Where $C[k]$ is defined as in the MOD convolution above.

Just like Isaac, you are also competing in the *Tercera Fecha del Gran Premio de México ICPC 2025*, and you should find the answer for this problem.

Input

The first line contains two integers n and m ($1 \le n, m \le 10^6$).

The second line contains n integers $A_1, \ldots, A_n$ ($0 \le A_i \le 10^6$).

The third line contains m integers $B_1, \ldots, B_m$ ($0 \le B_j \le 10^6$).

Output

Print a single integer — the answer to the problem.

Sample input 1	**Sample output 1**
2 2 5 3 6 4	20

Sample input 2	Sample output 2
3 4 3 4 8 2 0 2 5	197