D. Powerful array

time limit per test: 5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

An array of positive integers $a_1, a_2, ..., a_n$ is given. Let us consider its arbitrary subarray $a_l, a_{l+1}..., a_r$, where $1 \le l \le r \le n$. For every positive integer s denote by K_s the number of occurrences of s into the subarray.
We call the *power* of the subarray the sum of products $K_s \cdot K_s \cdot s$ for every positive integer s. The sum contains only finite number of nonzero summands as the number of different values in the array is indeed finite.

You should calculate the power of t given subarrays.

Input

First line contains two integers n and t ($1 \le n, t \le 200000$) — the array length and the number of queries correspondingly.

Second line contains n positive integers a_i ($1 \le a_i \le 10^6$) — the elements of the array.

Next t lines contain two positive integers l, r ($1 \le l \le r \le n$) each — the indices of the left and the right ends of the corresponding subarray.

Output

Output t lines, the i-th line of the output should contain single positive integer — the power of the i-th query subarray.

Please, do not use `%lld` specificator to read or write 64-bit integers in C++. It is preferred to use `cout` stream (also you may use `%I64d`).

Examples

input
3 2 1 2 1 1 2 1 3

output
3 6

input
8 3 1 1 2 2 1 3 1 1 2 7 1 6 2 7

output
20 20 20

Note

Consider the following array (see the second sample) and its [2, 7] subarray (elements of the subarray are colored):

Then $K_1 = 3, K_2 = 2, K_3 = 1$, so the power is equal to $3^2 \cdot 1 + 2^2 \cdot 2 + 1^2 \cdot 3 = 20$.