



□□□□□C/C++/Rust/Pascal 1□□□□□□2□

□□□□□C/C++/Rust/Pascal 512 M□□□□□1024 M

64bit IO Format: %lld

We define a special mysterious XOR operation \oplus_m with the following rules:

For two numbers a and b, first compute their regular XOR result $c=a\oplus b$. Then process c's binary representation as follows:

- 1. Scan c's binary representation from the least significant bit to the most significant bit
- 2. Initialize a counter count=0
- 3. For each bit:
 - If the bit is 1:
 - Increment count by 1
 - If count is odd, keep the bit
 - If count is even, clear the bit (set to 0)
 - · If the bit is 0, leave it unchanged

Example: $(101001)_2 \oplus_m (10010)_2 = (101001)_2$

Given an array A of length N, compute the sum of the mysterious XOR of A_i and A_j for all unordered pairs (i,j) where $i \neq j$. Formally, this can be expressed as $\sum_i^N \sum_{j>i}^N A_i \oplus_m A_j$.

0000:

The first line contains an integer N ($2 \leq N \leq 10^5$).

The second line contains N integers A_i ($0 \leq A_i \leq 10^8$).

Output a single integer representing the sum of mysterious XOR results for all unordered pairs.

① C++?clang++18?

1

ACMDE

ШШЦ