Task Xor

Fran recently learned the operation *xor*, which for two integers x and y returns the result by applying the bitwise exclusive or (*exclusive or*). The operation *xor*, denoted as $\oplus$, compares the corresponding bits of the numbers x and y and sets the result bit at each position according to the following rule:

- If the bits at the corresponding position are different (0 and 1, or 1 and 0), then the result bit is 1.
- If the bits are the same (0 and 0, or 1 and 1), then the result bit is 0.

For example, for $x = 5$ and $y = 3$, the binary representations are: $x = 101_2$, $y = 011_2$. Applying *xor* to the corresponding bits gives $x \oplus y = 101_2 \oplus 011_2 = 110_2 = 6$. In other words, $5 \oplus 3 = 6$.

Fran received an array of n integers $a_1, a_2, \ldots, a_n$ and decided to do the following:

1. For every pair of indices (i, j) where $1 \leq i \leq j \leq n$, he calculated the sum $a_i + a_j$.
2. Now he wants to calculate the result of the *xor* of all the obtained sums.

Help Fran calculate the required result.

Input

In the first line of input, there is n ($1 \leq n \leq 5 \cdot 10^5$), the length of the array.

In the second line, there are n numbers $a_1, a_2, \ldots, a_n$ ($0 \leq a_i < 2^{30}$) as described in the problem statement.

Output

In the only line of output, print the required result.

Scoring

Subtask	Points	Constraints
1	7	$n \leq 2000$
2	17	$a_i < 2^{10}$ for every i
3	45	$n \leq 10^5$
4	21	No additional constraints.

Examples

input	**input**	**input**
3 2 4 5	4 6 7 3 1	7 2 3 5 7 9 11 13
output	**output**	**output**
14	3	6

Clarification of the first example:
The sums are $2+2=4$, $2+4=6$, $2+5=7$, $4+4=8$, $4+5=9$, and $5+5=10$. The result is $4 \oplus 6 \oplus 7 \oplus 8 \oplus 9 \oplus 10 = 14$.