

---

## Problem Salt Lake City. Present

Input file: `input.txt` or standard input  
Output file: `output.txt` or standard output  
Time limit: 3 seconds  
Memory limit: 512 megabytes

Catherine received an array of integers as a gift for March 8. Eventually she grew bored with it, and she started calculating various useless characteristics for it. She succeeded to do it for each one she came up with. But when she came up with another one — **xor** of all pairwise sums of elements in the array, she realized that she couldn't compute it for a very large array, thus she asked for your help. Can you do it? Formally, you need to compute

$$\begin{aligned} &(a_1 + a_2) \oplus (a_1 + a_3) \oplus \dots \oplus (a_1 + a_n) \oplus \\ &\quad \oplus (a_2 + a_3) \oplus \dots \oplus (a_2 + a_n) \oplus \\ &\quad \quad \quad \dots \\ &\quad \quad \quad \oplus (a_{n-1} + a_n) \end{aligned}$$

### Input

The first line contains a single integer  $n$  ( $2 \leq n \leq 400\,000$ ) — the number of integers in the array.

The second line contains integers  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^7$ ).

### Output

Print a single integer — xor of all pairwise sums of integers in the given array.

### Examples

input	output
2 1 2	3
3 1 2 3	2

### Note

In the first sample case there is only one sum  $1 + 2 = 3$ .

In the second sample case there are three sums:  $1 + 2 = 3$ ,  $1 + 3 = 4$ ,  $2 + 3 = 5$ . In binary they are represented as  $011_2 \oplus 100_2 \oplus 101_2 = 010_2$ , thus the answer is 2.

$\oplus$  is the bitwise xor operation. To define  $x \oplus y$ , consider binary representations of integers  $x$  and  $y$ . We put the  $i$ -th bit of the result to be 1 when exactly one of the  $i$ -th bits of  $x$  and  $y$  is 1. Otherwise, the  $i$ -th bit of the result is put to be 0. For example,  $0101_2 \oplus 0011_2 = 0110_2$ .

---