Stone game

Alice and Bob are playing the game of Nim with *n* piles of stones (p[0], p[1], ..., p[n-1]). If Alice plays first, she loses if and only if the 'xor sum' (or 'Nim sum') of the piles is zero, i.e. p[0] ⊕ p[1] ⊕ ... p[n-1] = 0.

Bob can remove some stones in some piles before the game starts, but he must keep at least one pile unchanged. You task is to count the number of ways Bob can remove the stones to force Alice into losing the game. Since the number can be very large, output the number of ways modulo (% operator) 1000000007. Assume that both players will try to optimize their strategy and try to win the game.

Input

The first line of the input contains an integer 'n' denoting the number of piles. The next line contains space separated list of number of stones 'n' for each pile p[0] p[1]... p[n-1] respectively.

Output

An integer which is the 'number of ways'% 1000000007 Bob can force Alice to lose the game.

Constraint

3 <= n <= 100
0 < p[i] < 10^9

Sample Input

```
3
1 2 3
```

Sample Output

```
4
```

Explanation:

These are the possible changes:

- 0 2 2
- 1 0 1
- 1 1 0
- 1 2 3

For example, 1 ⊕ 2 ⊕ 3 = 0 so it will win.

However, (0 1 1) is wrong since he must keep one pile unchanged.

Scoring

Your score for this challenge will be based on the number of test cases your code passes.