

6678 Hack Protection

Pavel is sending to his friend Egor some array of non-negative integers. He wants to be sure, that nobody hacks the array before his friend gets it. To solve this problem Pavel need to compute some kind of a checksum or a digest for his array. Pavel has an innovative mind, so he invents the following algorithm to compute the digest for his array: count the number of subarrays in which the bitwise xor of the numbers in the subarray is equal to the bitwise and of the same numbers.

For example, consider an array of four binary numbers “01”, “10”, “11”, and “11”. The table below to the left lists the results of the bitwise xor of numbers for each subarray of this array, and the table below to the right list the results of the bitwise and of numbers for each subarray of this array. The rows of the table correspond to the starting elements of the subarray from the 1st element of the array to the 4th one, while columns correspond to the ending elements of the subarray. Matching values are highlighted with gray background.

01	11	00	11
	10	01	10
		11	00
			11

01	00	00	00
	10	10	10
		11	11
			11

Your task is to help Pavel compute this kind of a digest of the given array.

Input

The input file contains several test cases, each of them as described below.

The first line contains one integer n ($1 \leq n \leq 100000$). The second line contains n non-negative integers a_i ($0 \leq a_i \leq 2^{31} - 1$) that are written in decimal notation.

Output

The input file contains several test cases, each of them as described below.

On the first line of the output print Pavel’s digest of the given array.

Note: The sample input below corresponds to the example from the problem statement.

Sample Input

```
4
1 2 3 3
```

Sample Output

```
6
```