Powerful Subarrays

A subarray is said to be powerful if and only if the bit-wise xor of all its elements is a power of 2.

You are given an array of n integers $a_1, a_2, a_3 \dots a_n$. Can you count the number of powerful subarrays of the given array?

Consider, for example, a = [1, 2, 3]. The subarrays [1], [2], [2, 3] have xor-sum [1], [2] and [1] respectively which are perfect powers of 2. The count of powerful subarrays for [1, 2, 3] is 3.

Function Description

Complete the function *countPowerfulSubarrays* in the editor below. The function must return a single integer denoting the number of powerful subarrays.

countPowerfulSubarrays has one parameter -

arr: an integer array

Input Format

The first line of the input contains in integer, n, denoting the size of array.

Next line contains n space-separated positive integers $a_1, a_2, a_3 \dots a_n$ denoting the array elements.

Constraints

- $1 \le n \le 10^5$
- $1 \le a_i \le 10^9$

Output Format

Print the number of powerful subarrays. There is a code stub to handle I/O if you choose to use it.

Sample Input 0

```
5
1 2 3 4 5
```

Sample Output 0

```
8
```

Explanation 0

The powerful subarrays are:

$[1, 1]$, $[1, 4]$, $[1, 5]$, $[2, 2]$, $[2, 3]$, $[3, 5]$, $[4, 4]$, $[4, 5]$

Sample Input 1

```
3
1 3 9
```

Sample Output 1

```
2
```

Explanation 1

The powerful subarrays are:

$[1, 1]$, $[1, 2]$