STELELE INFORMATICII

Liceul Teoretic Internațional de Informatică București

Problem Caterpillar 3

Input file	`stdin`
Output file	`stdout`

— Sorry, it was a typo. We didn't mean to hurt your feelings like that.
— More often than not, we are stuck rationalizing whatever disreputable deeds we engage in for the sake of moving on. But, it is important in this process that we do not stray away from the one truth we have, which is the present moment.

In the quest of simplifying historical data for the sake of potential clarity, we can find ourselves destroying relationships and communities. What could be holier than the present, we ask of you? What is the point when mere abstention becomes problematic revisionism, and why shouldn't we believe you have crossed this threshold already?

– The Author with the Very Hungarian Catterpilar

Task

Consider the bitwise `AND` operation (hereby noted as `&`)[1].

We call a sequence b of numbers $b_1, b_2, \ldots, b_k$ *self-describing* if there exists an index p with $1 \leq p \leq k$ such that $b_1 \,\&\, b_2 \,\&\, \ldots \,\&\, b_k = b_p$.

Given a sequence v of N natural numbers $v_1, v_2, \ldots, v_N$, determine how many pairs of indexes (l, r) bound a *self-describing* subarray. That is, count the number of pairs of numbers (l, r) with with $1 \leq l \leq r \leq N$ such that $v_l, v_{l+1}, \ldots v_r$ is a *self-describing* sequence.

Input data

The first line contains a natural number N. The second line contains N natural numbers, representing the elements of the sequence v.

Output data

Print a single number — the sought number of pairs of indexes l, r that satisfy the property described in the statement.

Constraints

- $1 \leq N \leq 2\,000\,000$.
- $0 \leq v_i < 2^{60}$.

[1]The bitwise `AND` operation (denoted by `&`) is a binary operation that compares each bit of two operands and returns a new value where each bit is set to 1 only if the corresponding bits of both operands are 1. In other words, it performs a logical AND on every pair of corresponding bits. For example, given two 4-bit binary numbers $A = 1100_{(2)}$ and $B = 1010_{(2)}$, their bitwise AND is $A \,\&\, B = 1000_{(2)}$. This operation is defined in `C++` as `&`, which works similarly to any other operator (e.g. `+`, `-`, etc.)

#	Points	Constraints
1	10	$1 \leq N \leq 200$, $v_i < 2^{30}$
2	3	$1 \leq N \leq 2 \cdot 10^6$, all values v_i are equal
3	13	$1 \leq N \leq 2 \cdot 10^3$, $v_i < 2^{30}$
4	11	$1 \leq N \leq 4 \cdot 10^4$, $v_i < 2^{30}$
5	6	$1 \leq N \leq 2 \cdot 10^5$, $v_i = 2^k$
6	11	$1 \leq N \leq 2 \cdot 10^5$, $v_i < 8$
7	14	$1 \leq N \leq 2 \cdot 10^5$, $v_i = 2^j - 2^k$, for $0 \leq k \leq j \leq 60$
8	13	$1 \leq N \leq 2 \cdot 10^5$
9	19	No additional restrictions

Examples

stdin	stdout	Explanations
5 1 1 4 0 2	13	For the given example, the intervals that satisfy the property are: $(1,1)$, $(1,2)$, $(1,4)$, $(1,5)$, $(2,2)$, $(2,4)$, $(2,5)$, $(3,3)$, $(3,4)$, $(3,5)$, $(4,4)$, $(4,5)$, $(5,5)$.