MONSTERS

The 2017 International Congress of Monsters gathers *n* monsters coming from all over the world. Their chairman has to solve the following problem: if the i^{th} monster ($1 \le i \le n$) has k_i fingers, indexed from 0 to k_i - 1, so he can lift *j* of those fingers ($0 \le j \le k_i$), obtaining a certain number, in the following way: if a certain finger is lifted, $2^{\text{finger index}}$ is added to the current number. As a result, the i^{th} monster can count on his fingers nr_i distinct numbers. Therefore, the demanded result is $nr_1 + nr_2 + ... + nr_n$, modulo 10^9+7.

TASK

Compute the required sum, modulo 10^9+7.

INPUT FORMAT

The first line of the input file, ***monsters.in***, contains the number *n*.

The second line contains *n* positive integers, $k_1, k_2, ..., k_n$, representing the numbers of fingers of each monster.

OUTPUT FORMAT

The output file, ***monsters.out***, must contain a single positive integer, the requested sum, modulo 10^9+7.

LIMITS AND CONSTRAINTS

- $n \le 200{,}000$, $0 \le k_i \le 1{,}000{,}000{,}000$.
- The fingers are indexed from 0.

SUBSTASKS

Subtask	Score	Additional input constraints
1	40	$n \le 1.000$, $k_i \le 10.000$
2	100	$n \le 200.000$, $k_i \le 1.000.000.000$

EXAMPLE

monsters.in	*monsters.out*
2 3 7	136

EXPLANATIONS

The first monster can obtain 8 numbers:

0 - no finger was lifted;

1 - the index of lifted finger is 0;

2 - the index of lifted finger is 1;

3 - the indexes of lifted fingers are 0 and 1;

4 - the index of lifted finger is 2;
5 - the indexes of lifted fingers are 0 and 2;
6 - the indexes of lifted fingers are 1 and 2;
7 - the indexes of lifted fingers are 0, 1 and 2.

The second monster can obtain 128 numbers.