Descending Alternating Sums

Given an array **A** of **k** integers (not necessarily distinct), we define the *descending alternating sum* of this array, denoted **F(A)** the following way. First, we sort the array in descending order. Suppose the elements, after sorting, are $A_1 \ge A_2 \ge ... \ge A_k$ respectively. Then the descending alternating sum of array **A** is

$$F(A) = A_1 - A_2 + A_3 - ... + (-1)^{k+1} A_k.$$

For example, if **A = [5, -3, 8, 2, 0, -5]** then after sorting it in descending order, we find **A = [8, 5, 2, 0, -3, -5]**. So the descending alternating sum of this array is **8 - 5 + 2 - 0 + (-3) - (-5) = 7**. In particular, the descending alternating sum of an empty array is **0**.

You are given an array **A** of **n** integers where $1 \le n \le 10^5$ and $|A_i| \le 10^{18}$. You have to print the sum of the descending alternating sums of all subsets of this array **A** (there are 2^n of them) modulo $M = 10^9 + 7$. In other words, if the subsets of array **A** are $S_1, S_2, ..., S_{2^n}$ then you have to print the sum

$$F(S_1) + F(S_2) + ... + F(S_{2^n}) \text{ modulo } M = 10^9 + 7.$$

Note: we consider some integer modulo a positive integer to be non-negative. In the other words, the output **R** must satisfy the inequality $0 \le R < M$.

Input

The first line of the input file contains a single integer **n**, denoting the size of the array **A**.

The second line contains **n** integers $A_1, A_2, ..., A_n$, the elements of the array **A**.

Output

Print a single integer, the sum of descending alternating sums of all subsets of the array **A**.

Example

Input:
```
3
-1 9 3
```

Output:
```
36
```