Subset Fight (subsetfight)

Marco is playing a game using a deck of cards. The deck consists of K different types of cards. For each type $i = 1 \dots K$ there are V_i cards of that type, which are distinguishable by their seed, but they all share the same *value* i. Notice the total number of cards in the deck is $N = V_1 + V_2 + \cdots + V_K$. For example, let's say that $K = 3$ and $V = [1, 2, 3]$. This means that the values of the cards in the deck are as follows:

$$1, 2, 2, 3, 3, 3$$

Marco must choose anyone of the possible 2^N subsets of cards, including the empty subset with zero cards. If the sum of the values in the subset is a multiple of K, Marco wins! In the example above, there are 24 different ways of choosing a winning subset of cards.

Figure 1: A possible subset of cards chosen by Marco.

Since the game is quite easy, Marco is now wondering: **how many ways** are there to win the game?

> ☞ Among the attachments of this task you may find a template file `subsetfight.*` with a sample incomplete implementation.

Input

The first line contains the only integer K. The second line contains K integers V_i.

Output

You need to write a single line with an integer: the number of different winning subsets of cards **modulo** $10^9 + 7$ (cards with the same value but different seed are considered different).

✍ The *modulo* operation (a mod m) can be written in C/C++/Python as `(a % m)` and in Pascal as `(a mod m)`. To avoid the *integer overflow* error, remember to reduce all partial results through the modulus, and not just the final result!
Notice that if $x < 10^9 + 7$, *then* $2x$ *fits into a* C/C++ `int` *and* Pascal `longint`.

Constraints

- $1 \leq K \leq 100$.
- $1 \leq V_i \leq 200\,000$ for each $i = 1 \ldots K$.

Scoring

Your program will be tested against several test cases grouped in subtasks. In order to obtain the score of a subtask, your program needs to correctly solve all of its test cases.

– **Subtask 1** (0 points) Examples.

– **Subtask 2** (20 points) The total number of cards in the deck is at most 20.

– **Subtask 3** (40 points) The total number of cards in the deck is at most 100 000.

– **Subtask 4** (40 points) No additional limitations.

Examples

input	output
3 1 1 1	4
5 69 420 1017 128 953	985611225

Explanation

In the **first sample case** the correct subsets are:

- The empty set $\{\}$,
- The two cards of values $\{1, 2\}$,
- The only card of value $\{3\}$,
- The three cards of values $\{1, 2, 3\}$.