Problem A. Learning to Add

Input file:	`Standard input`
Output file:	`Standard output`
Time limit:	2 seconds
Memory limit:	256 mebibytes

Little Tomek is learning how to add. His teacher wrote out a sequence of n numbers and Tomek adds them one by one (in order) to his result. At the beginning, the result is equal to zero. However, from time to time, he mixes up the order of digits. In particular, after performing an addition, he writes down the new result with randomly permuted digits, discarding any leading zeros. For example, if the previous result was 67 and Tomek added 42, the new result could be 109, 901, 190, 910, 91 or 19.

What is the maximum result Tomek can obtain after n additions?

Input

The first line of input contains an integer n ($1 \leq n \leq 5$) – the length of the sequence Tomek has to sum. The second line contains n integers $a_1, \ldots, a_n$ ($1 \leq a_i \leq 100$) – the elements of the sequence, in order.

Output

Output the maximum result Tomek can obtain.

Examples

Standard input	Standard output

`5`
`42 1 3 3 6`
	`100`

Note

Tomek could perform the following additions, in order:

- $0 + 42 = 42$
- $42 + 1 = 43$
- $43 + 3 = 46$
- $46 + 3 = \mathbf{94}$
- $94 + 6 = 100$

Note that here Tomek makes only one mistake; in general, he may make any number of mistakes.

Problem B. Adjusting Ducks

Input file:	`Standard input`
Output file:	`Standard output`
Time limit:	2 seconds
Memory limit:	256 mebibytes

Wojtek's ducks' constant quacking annoys Tomek a great deal. When approached about the issue, Wojtek told Tomek to take it easy... Things got serious when Tomek bought a sling. They held a diplomatic meeting to reach a compromise. It turns out that when two or more ducks have the same quacking pitch, the sound becomes so constant, that they do not disturb Tomek at all. Fortunately enough, the local veterinarian can alter the pitches of ducks. For changing a duck's quacking pitch from a to b, he demands $|b - a|$ zlotys. How many zlotys does Wojtek have to spend to make it so no duck is alone at its pitch?

Input

The first line of the input contains an integer n ($2 \le n \le 10^5$), the number of ducks. The second line contains n integers $a_1, \ldots, a_n$ ($1 \le a_i \le 10^8$), the pitches of the ducks.

Output

Output the minimal cost of adjusting the ducks' pitches so that Tomek is no longer annoyed.

Examples

Standard input	Standard output
5 42 1 3 3 6	38

Note

One possible solution is to adjust the pitch of the second duck to 3 (for a price of 2) and the pitch of the last duck to 42 (for a price of 36).

Problem C.

Input file:	`Standard input`
Output file:	`Standard output`
Time limit:	2 seconds
Memory limit:	256 mebibytes

Tomek has found a pile of sticks and now plans to build a polygon with four of them as its sides. Wojtek, however, insists that the quadrilateral they construct be concave. That is, it should have at least one internal angle strictly greater than 180 degrees. Moreover, no two sides of the quadrilateral can touch in points other than the vertices.

What is the largest possible perimeter of the concave quadrilateral that Tomek and Wojtek construct?

Input

The first line of input contains an integer n $(1 \le n \le 10\,000)$ – the number of sticks found by Tomek. The second line contains n integers $a_1, \ldots, a_n$ $(1 \le a_i \le 10^8)$ – the lengths of Tomek's sticks.

Output

Output the maximum possible perimeter of a concave quadrilateral that can be constructed using four of the given sticks. If there is no quadrilateral that satisfies the requirements, output -1.

Examples

Standard input	Standard output
5 42 1 3 3 6	13

Note

For the above example, the concave quadrilateral with maximum perimeter is constructed using the last four sticks.

Problem D. Splitting Money

Input file:	`Standard input`
Output file:	`Standard output`
Time limit:	2 seconds
Memory limit:	256 mebibytes

While digging a well, Tomek and Wojtek found a jar full of old bills. Even though some of them may had the same nominals, they were all fully distinguishable. They were not old enough to be worth any actual money though, so the guys decided to split them in quite an extraordinary way. Tomek will take some of the bills (maybe none), then Wojtek will take some (maybe none), so that the *xor* of Wojtek's nominals equals the *xor* of Tomek's. The rest will be left in the jar. How many ways in which this can be done are possible?

Input

The first line of the input contains an integer n $(1 \leq n \leq 10\,000)$, the number of bills. The second line contains n integers $a_1, \ldots, a_n$ $(1 \leq a_i \leq 10\,000)$, the nominals of the bills.

Output

Output the number of money splits such that the *xor* of Wojtek's nominals equals the *xor* of Tomek's. The answer might be big, so output it modulo $10^9 + 7$.

Examples

Standard input	Standard output

`5`
`42 1 3 3 6`
	`5`

Note

In the above example, the possible splits are:

- $\{3\} : \{3\}$ (two ways)
- $\{3, 3\} : \{\}$
- $\{\} : \{3, 3\}$
- $\{\} : \{\}$

Problem E. Burger Bar

Input file:	`Standard input`
Output file:	`Standard output`
Time limit:	2 seconds
Memory limit:	256 mebibytes

In addition to being a duck breeder, Wojtek manages a Burger Bar. It is said to be the best in town, or at least one of the best. Well, for sure it is a pretty good place. Not the worst, by any means. People do like eating there. At least Tomek does. He was recommending Wojtek's bar to some of his friends lately, but they wanted to know how many toppings are there to choose from. Tomek did not remember, but fortunately he had some of the receipts in his wallet. Every time he bought a burger, he chose some subset of the available toppings and got billed for the sum of their prices. Knowing how many zlotys Tomek paid for the toppings every time, calculate the minimum possible number of toppings in Wojtek's restaurant.

Input

The first line of the input contains an integer n ($1 \le n \le 20$), the number of receipts. The second line contains n integers $a_1, \ldots, a_n$ ($1 \le a_i \le 50$), the costs of Tomek's toppings on the receipts.

Output

Output the minimum possible number of toppings in Wojtek's restaurant. Note that price of each topping in Wojtek's bar is non-negative integer.

Examples

Standard input	Standard output
5 42 1 3 3 6	4

Note

One of the many possible solutions is one kind of topping for 1 zloty, two kinds of toppings for 3 zlotys each, and one kind of topping for 35 zlotys.

Problem F. Data Mining

Input file:	`Standard input`
Output file:	`Standard output`
Time limit:	2 seconds
Memory limit:	256 mebibytes

Wojtek is a programmer at Bytes, Inc. He is charged with collecting data, and has a cluster of n serially connected servers at his disposal. Server 1 is connected only to server 2, server 2 is connected to servers 1 and 3, and so on, up to server n, connected only to server $n-1$. Wojtek just learned the upper management is considering cutting funding to his team. In particular, they are going to claim all the servers with low amount of storage used.

Wojtek would like to prevent this from happening, so he resolved to fill his servers with data. To do that, he can pick any server, and copy all data on it to one of the servers it is connected to. Wojtek has enough time to perform this operation k times before the management makes a decision. What is the maximum possible least amount of storage used on any of the servers after Wojtek performs k copy operations?

To clarify: let the amounts of storage used on the n servers be $s_1, \ldots, s_n$ after Wojtek is done. We are interested in maximizing $\min(s_1, \ldots, s_n)$. You can assume that the free storage space available on all the servers is effectively unbounded.

Input

The first line of input contains the integer k ($0 \le k \le 10^6$). The second line contains the integer n ($2 \le n \le 10\,000$). The third line contains n integers $a_1, \ldots, a_n$ ($1 \le a_i \le 10^8$) – the initial amounts of storage used on the servers.

Output

Output the maximum possible least amount of storage used on any of the servers after Wojtek performs k copy operations. Because the answer can be very big, output it modulo 10^9+7.

Examples

Standard input	Standard output

`5`
`5`
`42 1 3 3 6`
	`43`

Note

For the above example, Wojtek can perform the following sequence of copy operations:

- server 2 to server 1,
- server 1 to server 2,
- server 2 to server 3,
- server 3 to server 4,
- server 4 to server 5.

The final amounts of storage used on each server are $43, 44, 47, 50$ and 56, respectively.