Lightweight Ladder (`ladder`)

In the recent festivities, Luca observed some kids playing in a gym. They seemed to enjoy very much moving around with all these colorful blocks, but sometimes they were unable to climb on them as they were too high when stacked!

Figure 1: Kids at work.

Consider this simplified representation of a typical situation: each block is a cube with all sides of one meter. Some cubes can optionally be stacked to form a heap and the heaps are aligned one next to each other, as in picture 2.

When two heaps have the same number of cubes stacked, a kid can just walk to proceed to the next one. Similarly, in case of a descent when the next heap has fewer cubes than the current one, the kid exploits the gravity and falls gently on the soft cube.

As gravity does not work in the reverse direction, kids cannot climb to higher heaps without the help of a ladder. In the situation presented in figure 2, two ladders can be strategically placed between the start and the first heap and before the last heap.

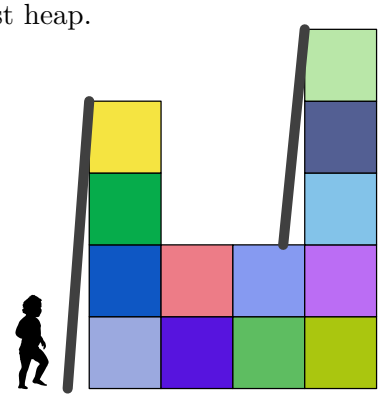


Figure 2: A sequence of four heaps with two ladders of length 4 and 3 meters.

Placing these ladders all over the path can be quite expensive, especially when it is really long. Luca had an idea: kids could carry themselves a single lightweight ladder to reach the end of the path; thus, they need a ladder which covers all their needs, optionally being longer in some cases, but never shorter (otherwise they cannot make it!).

What is the shortest ladder that one can carry to complete the path from the ground at the beginning until the last of N heaps?

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

Input

The first line contains the number N of heaps. The second line contains N integers C_i, the number of cubes stacked in the i-th heap.

Output

You need to write a single line with an integer: the minimum length of a ladder that can be carried to complete the path.

Constraints

- $1 \leq N \leq 1\,000\,000$.
- $0 \leq C_i \leq 10^9$ for each $i = 0 \ldots N-1$.
- In case no ladder is needed at all, you must output the value 0.

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** (10 points) $N \leq 2$.

– **Subtask 3** (45 points) $N, C_i \leq 1000$ for each $i = 0 \ldots N-1$.

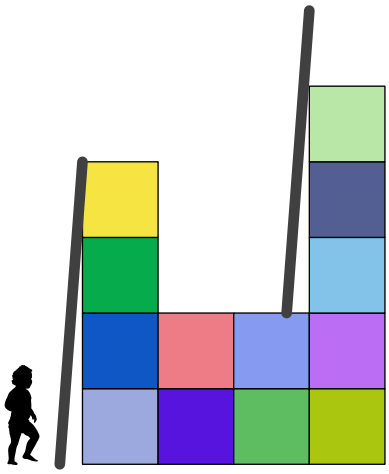
– **Subtask 4** (45 points) No additional limitations.

Examples

input	output
4 4 2 2 5	4
10 2 6 3 7 0 0 7 3 6 4	7

Explanation

In the **first sample case**, already described in this statement, the shortest possible length for the ladder is 4 meters (note that in the picture they are shown as two different ladders, but in reality it is just one carried by the kid as he proceeds):



In the **second sample case** we need at least a ladder of length 7 meters (any higher length would work as well, but we are required to find the minimum solution):