

# Band for Band

time limit per test: 3 seconds (3 seconds for Python)  
memory limit per test: 500 megabytes  
input: standard input  
output: standard output  
points: 100 · problem code: 1008sorting1

---

The TAs of FIT1008 are holding an undercover tournament that they are calling "Band for Band".

In the tournament, every competitor starts off with a certain number of tokens. Then, two competitors are repeatedly chosen at random. These competitors then go "Band for Band".

The competitor who currently has more tokens wins and receives all of the other competitor's tokens. If both competitors have the same number of tokens, then the winner is chosen randomly with a coin flip (and they get the other competitor's tokens). The losing competitor is then removed from the tournament. This continues until only one competitor stands, holding everyone's tokens.

For example, if there are 4 competitors A, B, C and D, and they have [1, 2, 4, 3] tokens respectively, then one possible outcome of the tournament (of many) is:

- The first game is between A and D. D has more tokens, so they win (D now has 4 tokens).
- The second game is between C and D. They have an equal number of tokens. Suppose C wins the coin flip (C now has 8 tokens).
- The third and final game therefore is between B and C. C has more tokens, so they win (C now has 10 tokens).
- C is the last survivor and the winner of the tournament.

You know ahead of time how many tokens every competitor has. Your task is to find how many competitors have a non-zero chance of winning the whole tournament.

In order to pass the tests, your solution should have a time complexity of  $O(n \log(n))$ .

## Input

---

Input will begin with a single integer  $n$  ( $1 \leq n \leq 10^5$ ) — the number of competitors. The next line will contain  $n$  integers  $a_i$  ( $1 \leq a_i \leq 10^9$ ), where  $a_i$  is the number of tokens the  $i$ -th competitor has.

## Output

---

Print a single integer — the number of competitors that have a non-zero chance of winning the whole tournament.

## Example 1

---

### Input

```
4
1 2 4 3
```

### Output

```
3
```

In this scenario, the 2nd, 3rd and 4th competitors can possibly win the tournament. It is impossible for the 1st competitor to win the tournament (they will always lose the first game they play).

## Example 2

---

### Input

```
5
1 1 1 1 1
```

## Output

5

Anyone can win this tournament.