

## Library of Binaria (binaria)

The *Library of Binaria* faced some organizational challenges, so they created a new set of ‘Rules’ for storing books. They have large shelves with ample space above them, and the books are categorized by *topic*. Each book belongs to exactly one topic. To follow these new ‘Rules’, they want to organize the books on the shelves as follows:

- Books belonging to the same topic must be on the same shelf.
- Books belonging to different topics must be on different shelves.
- Books within one shelf should be arranged in rows where the number of books doubles with each successive row: the first row contains one book, the second row contains two books, the third row contains four, the fourth row contains eight, and so on.



You are provided with the total number of books for each topic. The goal is to determine whether the books can be packed according to these ‘Rules’. If it is impossible, calculate the minimum number of additional books the library has to buy to achieve proper storage. You don’t need to break down the number of books per topic – it is enough to provide the overall sum.



Figure 1: A bookshelf in the Library of Binaria before the ‘Rules’.

Create a program to assist Chen-Ben Chuchul, the librarian of Binaria, determining the total number of books the library has to order to properly organize their collection on the shelves.

📎 Among the attachments of this task you may find a template file `binaria.*` with a sample incomplete implementation.

## Input

The first line contains an integer  $N$ , the number of topics. The second line contains  $N$  space separated integers  $T_i$ , the number of books belonging to each topic in the library of Binaria.

## Output

You need to write a single line with an integer: the total number of books to be ordered by the library.

## Constraints

- $1 \leq N \leq 100\,000$ .
- $1 \leq T_i \leq 10^{13}$  for each  $i = 0 \dots N - 1$ .

## 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** (4 points)       $N \leq 100$ .  $T_i \leq 100$  for each  $i = 0 \dots N - 1$ .  
📖📖📖📖
- **Subtask 3** (32 points)       $N \leq 1000$ .  $T_i \leq 10^6$  for each  $i = 0 \dots N - 1$ .  
📖📖📖📖
- **Subtask 4** (64 points)      No additional limitations.  
📖📖📖📖

## Examples

input	output
3 8 3 25	13
2 7 1	0
2 100000000000000 100000000000000	7691697672190

## Explanation

In the **first sample case**:

- For the first topic we can only store 7 books in the first 3 ( $1 + 2 + 4 = 7 < 8$ ), and in 4 rows we can store  $1 + 2 + 4 + 8 = 15 > 8$  books. We need to buy  $15 - 8 = 7$  additional books.

- For the second topic we do not need to buy extra books as 3 books can be arranged in two rows ( $1 + 2 = 3$ ).
- For the third topic 4 rows is not enough as  $1+2+4+8 < 25$ . But 5 rows is enough as  $1+2+4+8+16 = 31 > 25$ . We need to buy  $31 - 25 = 6$  books.

So we need to buy  $7 + 0 + 6 = 13$  books in total.

In the **second sample case** we do not need to buy any books, as 7 books can be arranged in 3 rows ( $1 + 2 + 4 = 7$ ), and for 1 book we need only 1 row.