

## Problem : 3

Call an array **a** of length  $n$  sorted if  $a_1 \leq a_2 \leq \dots \leq a_{n-1} \leq a_n$

Rafiq has an array **a** of length  $n$

He is allowed to perform one type of operation on it (zero or more times):

- Choose an index  $i$  ( $1 \leq i \leq n-1$ ).
- Add 1 to  $a_1, a_2, \dots, a_i$
- Subtract 1 from  $a_i + 1, a_i + 2, \dots, a_n$

The values of **a** can be negative after an operation.

Determine the minimum operations needed to make **a** not sorted.

Input contains  $n$  integers  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^9$ ) — the values of array **a**

Output the minimum number of operations needed to make the array not sorted.

সারাংশঃ একটি সর্টেড array দেওয়া থাকবে, সেটাকে উপরের শর্ত অনুযায়ী আনসরটেড করতে মিনিমাম কতোগুলো অপারেশন করতে হবে।

Array will be like : (Consider as input)

1 1

1 8 10 13

1 3 2

1 9 14

**Output** : Log the minimum operation to make it unsorted. If the array is already unsorted log 0.

1

2

0

3

### Note

In the first case, we can perform 1 operation to make the array not sorted:

- Pick  $i=1$ . The array **a** then becomes  $[2,0]$ , which is not sorted.

In the second case, we can perform 2 operations to make the array not sorted:

- Pick  $i=3$ . The array **a** then becomes  $[2,9,11,12]$ .
- Pick  $i=3$ . The array **a** then becomes  $[3,10,12,11]$ , which is not sorted.

**It can be proven that 1 and 2 operations are the minimal numbers of operations in the first and second test cases, respectively.**

In the third case, the array is already not sorted, so we perform 0 operations.