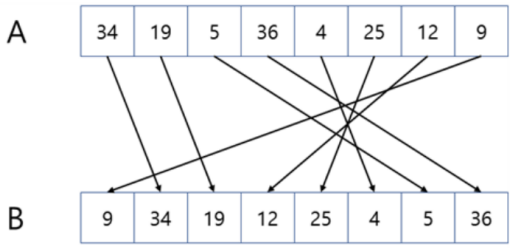
# Problem I. Integer Array Shuffle

| | |
|---|---|
| Input file: | *standard input* |
| Output file: | *standard output* |
| Time limit: | 1 second |
| Memory limit: | 1024 mebibytes |

Given an integer array $A$ of size $N$, the *shuffle* operation is defined as follows.

- Initially, you create an empty integer array $B$.
- Then, while $A$ is not empty, you remove either the leftmost or rightmost element of $A$, and append the value to the right in $B$.
- If $A$ is empty, replace $A$ with $B$ and stop.




If the shuffle operation is performed as shown in the picture above, the value of the array $A$ is changed as follows:

$$(34, 19, 5, 36, 4, 25, 12, 9) \to (9, 34, 19, 12, 25, 4, 5, 36).$$

Let $A_i$ be the $i$-th element of array $A$. When the condition "if $1 \le i < j \le N$, then $A_i \le A_j$" is established, it is said that array $A$ increases monotonically.

Write a program that, given an integer array $A$ of size $N$, calculates the minimum number of shuffle operations required to make the array $A$ monotonically increasing.

## Input

The first line of input contains one integer $N$, the number of elements in array $A$ ($1 \le N \le 3 \cdot 10^5$).

The second line contains $N$ integers $A_1, \ldots, A_N$: the initial values of elements of array $A$ ($1 \le A_i \le 10^9$).

## Output

Output the minimum number of shuffle operations required to make the array $A$ monotonically increasing.

## Examples

| standard input | standard output |
|---|---|
| 3<br>2 2 5 | 0 |
| 6<br>1 5 8 10 3 2 | 1 |