Consider the following sorting algorithm:

```
reverse-sort(sequence a)
    while (a is not in nondecreasing order)
        partition a into the minimum number of slopes
        for every slope with length greater than one
            reverse(slope)
```

A slope is defined as a decreasing consecutive subsequence of **a**. The reverse procedure will reverse the order of the elements in a slope.

You are given a permutation of the first **N** natural numbers whose slopes all have even length when partitioned for the first time. Determine the total number of times reverse is called to reverse-sort the given permutation.

INPUT

The first line of input contains the positive integer **N** ($2 \leq \mathbf{N} \leq 100\,000$).

The second line of input contains a permutation of the first **N** natural numbers that needs to be sorted.

OUTPUT

The only line of output must contain the number of times that reverse is called.

SAMPLE TESTS

input	input	input
2 2 1	4 4 3 2 1	4 3 1 4 2
output	output	output
1	1	3