

Problem C. StalinSort Algorithm

Input file: *standard input*
Output file: *standard output*
Time limit: 3 seconds
Memory limit: 512 mebibytes

There is an esoteric sorting *algorithm* called StalinSort. It goes as follows.

Go through the elements of the given array from left to right, starting from the second. If the current element is not less than previous, then do nothing, otherwise erase it. In the end you get a sorted array.

One can implement StalinSort in $O(n)$ time, which is pretty cool. There is a catch though: you can lose many elements in the process. To fix this I've come up with an improved nondeterministic version of this algorithm.

Go through the elements of the given array from left to right, starting from the second. If the current element is not less than previous, then do nothing, otherwise you have options. You can either erase it, or erase the previous element, but only if the current prefix still becomes non-decreasing. In the end you get a sorted array.

Depending on its choices, this algorithm can erase different number of elements. I wonder, what is the minimum number of erased elements I can get applying this improved version of StalinSort to the given permutation?

Input

The first line contains one positive integer n ($1 \leq n \leq 5 \cdot 10^5$) — the size of the permutation.

The second line contains n integers p_i ($1 \leq p_i \leq n$) — the permutation itself. It is guaranteed that all p_i are distinct.

Output

Print one number — the minimum number of elements improved StalinSort can erase.

Examples

standard input	standard output
6 6 1 2 3 4 5	1
6 5 6 1 2 3 4	4
6 6 4 5 1 2 3	3