Task: PRA
Diligent Johny

Olimpiada Informatyczna

XXIII OI, Stage III, Day 1. Source file `pra.*` Available memory: 128 MB. *13.04.2016*

It is Little Johny's birthday... And this is serious algorithmic problem, so the poor kid received no toys, games or a computer for his birthday present. Rather, he was presented with long arrays filled with numbers, trees, maps of strange lands rife with roads that lead through numerous tunnels and overpasses, lengthy tapes filled with 1048576 symbols long prefixes of Fibonacci and Thue-Morse words, etc. Of all these educational gifts, he likes an array holding a permutation* of the first n positive integers the most. Soon, Johny started wondering what is the lexicographic predecessor permutation† of the one he was given. Having figured that out rather quickly, Johny immediately asked himself how could he write this predecessor permutation in his array. The only operation that the array supports is swapping the contents of two arbitrary cells. Fortunately, Johny was smart enough to transform the initial permutation into its predecessor in the minimum number of swaps. He found this task so captivating, that he kept on transforming each successive permutation into its predecessor.

In his permutation madness, Johny is ignoring all his birthday party guests, which they find amusing enough but also a little rude. One of them soon realized that Johny will stop once he gets down to the identity permutation $1, 2, \ldots, n$, which is lexicographically smallest. The question is, how long will this take? Help them answer this question, knowing that every swap takes Johny exactly one second. As this might take a while (diligent is Johny's middle name), the guests will be happy enough to know the remainder of division by $10^9 + 7$. After all, they can check back on Johny every $10^9 + 7$ seconds to see if he is finally done.

Input

In the first line of the standard input, there is a single positive integer n, specifying the length of the permutation that Johny got for his birthday. In the second line, the permutation itself is given, as a sequence of n pairwise different integers $p_1, p_2, \ldots, p_n$ $(1 \leq p_i \leq n)$, separated by single spaces.

Output

Your program should print to the standard output the remainder of division by $10^9 + 7$ of the number of swaps that Johny will make before he stops.

Example

For the input data:	the correct result is:
`3` `3 1 2`	`6`

Explanation of the example: The lexicographically decreasing sequence of permutations that Johny will go through is $(2, 3, 1)$, $(2, 1, 3)$, $(1, 3, 2)$, $(1, 2, 3)$. To obtain those, he will make $2 + 1 + 2 + 1 = 6$ swaps in total.

Sample Grading Tests:

`1ocen:` $1, 2, 3, \ldots, 10$

`2ocen:` a random 5-element permutation

`3ocen:` $100, 99, 98, \ldots, 1$.

Grading

The set of tests consists of the following subsets. Within each subset, there may be several test groups.

*A permutation of the numbers from 1 to n is a sequence of pairwise different integers $p_1, \ldots, p_n$ satisfying $1 \leq p_i \leq n$ (i.e., every integer from 1 to n appears exactly once in a permutation).

†The permutation $P = (p_1, \ldots, p_n)$ is lexicographically smaller than the permutation $Q = (q_1, \ldots, q_n)$ (which we denote $P < Q$) if $p_j < q_j$, where j is the smallest index such that $p_j \neq q_j$. The permutation P is the lexicographic predecessor of Q if $P < Q$ and there exists no permutation R such that $P < R < Q$.

Subset	Property	Score
1	$n \leq 10$	15
2	$n \leq 5000$	37
3	$n \leq 1\,000\,000$, the permutation is $n, n-1, \ldots, 1$	15
4	$n \leq 1\,000\,000$	33