

Problem J. Sort It!

Input file: *standard input*
Output file: *standard output*
Time limit: 1.5 seconds
Memory limit: 512 mebibytes

You are given a permutation of length n : $p_1, p_2, \dots, p_n$. Consider some array of length n consisting of integers between 1 and n (equal elements are allowed). Let us transform the array in the following manner: at first, let us take all elements equal to p_1 from it and write them on a piece of paper (if there are no such elements, just do not write anything). Then write all elements equal to p_2 , then equal to p_3 and so on, finishing by all elements equal to p_n , thus obtaining a new array of length n . For example, if the permutation is 2 1 3 and the array is 2 3 2, the resulting array will be 2 2 3. If after this transformation we get a sorted array, let us call the original array *sortable by p* . Calculate the total number of arrays that are sortable by p .

As the answer can be very large, output it modulo $10^9 + 7$.

Input

The first line contains a single integer n ($1 \leq n \leq 2000$): the length of the permutation. The second line contains n distinct integers $p_1, p_2, \dots, p_n$ ($1 \leq p_i \leq n$): the permutation itself.

Output

Output a single integer: the answer to the problem modulo $10^9 + 7$.

Examples

standard input	standard output
2 2 1	2
3 2 1 3	15