E. Iahub and Permutations

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Iahub is so happy about inventing bubble sort graphs that he's staying all day long at the office and writing permutations. Iahubina is angry that she is no more important for Iahub. When Iahub goes away, Iahubina comes to his office and sabotage his research work.

The girl finds an important permutation for the research. The permutation contains n distinct integers $a_1, a_2, ..., a_n$ $(1 \le a_i \le n)$. She replaces some of permutation elements with -1 value as a revenge.

When Iahub finds out his important permutation is broken, he tries to recover it. The only thing he remembers about the permutation is it didn't have any fixed point. A fixed point for a permutation is an element a_k which has value equal to k $(a_k = k)$. Your job is to proof to Iahub that trying to recover it is not a good idea. Output the number of permutations which could be originally Iahub's important permutation, modulo 1000000007 $(10^9 + 7)$.

Input

The first line contains integer n $(2 \le n \le 2000)$. On the second line, there are n integers, representing Iahub's important permutation after Iahubina replaces some values with -1.

It's guaranteed that there are no fixed points in the given permutation. Also, the given sequence contains at least two numbers -1 and each positive number occurs in the sequence at most once. It's guaranteed that there is at least one suitable permutation.

Output

Output a single integer, the number of ways Iahub could recover his permutation, modulo 1000000007 $(10^9 + 7)$.

Examples

input
5 -1 -1 4 3 -1
output
2

Note

For the first test example there are two permutations with no fixed points are [2, 5, 4, 3, 1] and [5, 1, 4, 3, 2]. Any other permutation would have at least one fixed point.