

D. Permutations

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

A permutation is a sequence of integers from 1 to n of length n containing each number exactly once. For example, (1), (4, 3, 5, 1, 2), (3, 2, 1) are permutations, and (1, 1), (4, 3, 1), (2, 3, 4) are not.

There are many tasks on permutations. Today you are going to solve one of them. Let's imagine that somebody took several permutations (perhaps, with a different number of elements), wrote them down consecutively as one array and then shuffled the resulting array. The task is to restore the initial permutations if it is possible.

Input

The first line contains an integer n ($1 \leq n \leq 10^5$). The next line contains the mixed array of n integers, divided with a single space. The numbers in the array are from 1 to 10^5 .

Output

If this array can be split into several permutations so that every element of the array belongs to exactly one permutation, print in the first line the number of permutations. The second line should contain n numbers, corresponding to the elements of the given array. If the i -th element belongs to the first permutation, the i -th number should be 1, if it belongs to the second one, then its number should be 2 and so on. The order of the permutations' numbering is free.

If several solutions are possible, print any one of them. If there's no solution, print in the first line - 1.

Examples

input
9 1 2 3 1 2 1 4 2 5
output
3 3 1 2 1 2 2 2 3 2
input
4 4 3 2 1
output
1 1 1 1 1
input
4 1 2 2 3
output
-1

Note

In the first sample test the array is split into three permutations: (2, 1), (3, 2, 1, 4, 5), (1, 2). The first permutation is formed by the second and the fourth elements of the array, the second one — by the third, the fifth, the sixth, the seventh and the ninth elements, the third one — by the first and the eighth elements. Clearly, there are other splitting variants possible.