Problem F. Fragmentation

Input file:	`fragmentation.in`
Output file:	`fragmentation.out`
Time limit:	2 seconds
Memory limit:	256 megabytes

Felix is working on a startup project in his garage. He has already found a great name for his project: SuperFastZilla. By now he is not sure what SuperFastZilla should do, but he is pretty sure it should do it fast, super fast.

Once he noticed that SuperFastZilla is working too slow, inspite of the fast algorithms used in it. Felix thinks that the problem may be caused by storage fragmentation.

The storage used by SuperFastZilla consists of n blocks of memory. SuperFastZilla performs some operations on this storage. Each block is used in one operation only, the i-th block is used in the a_i-th operation.

Felix wants to sort these blocks by the index of the operation they are used. To make it faster, Felix wants to split the storage into minimal number of segments of consecutive blocks, and then rearrange these segments to get the sorted array of blocks. After this rearrangement the order of block's indices of operations must be non-decreasing.

Help Felix to find the way to split the storage that minimizes the number of segments.

For example, if $a = [2, 3, 1, 1, 2, 2, 1]$, it can be split into three parts: $[2, 3]$, $[1, 1, 2, 2]$ and $[1]$. These parts can be rearranged to make the sorted array: $[1]$, $[1, 1, 2, 2]$, $[2, 3]$.

Input

The first line of input file contains an integer n ($1 \le n \le 10^5$). The next line contains n integers a_i ($1 \le a_i \le 10^5$).

Output

The first line of the output file must contain an integer number m — the minimal number of segments.

The next line must contains m integers, the lengths of the segments, from left to right.

Example

fragmentation.in	fragmentation.out
7 2 3 1 1 2 2 1	3 2 4 1