Problem E. Increasing Subsequence

Time Limit	1000 ms
Mem Limit	1048576 kB
OS	Linux

A strictly increasing sequence is a sequence of numbers $a_1, a_2, \ldots, a_n$ such that, for $1 < i \leq n$, $a_{i-1} < a_i$. A subsequence of $a_1, a_2, \ldots, a_n$ is identified by a strictly increasing sequence of indices, $x_1, x_2, \ldots, x_m$ where $1 \leq x_1$ and $x_m \leq n$. We say $a_{x_1}, a_{x_2}, \ldots, a_{x_m}$ is a subsequence of $a_1, a_2, \ldots, a_n$. For example, given the sequence $8, 90, 4, 10\,000, 2, 18, 60, 172, 99$, we can say that $90, 4, 10\,000, 18$ is a subsequence but $8, 90, 18, 2, 60$ is not. The subsequence $4, 18, 60, 172$ is a subsequence that is, itself, strictly increasing.

Given a sequence of numbers, can you write a program to find a strictly increasing subsequence that is as long as possible?

Input

Input has up to 200 test cases, one per line. Each test case starts with an integer $1 \leq n \leq 200$, followed by n integer values, all in the range $[0, 10^8]$. A value of zero for n marks the end of input.

Output

For each test case, output the length of the longest strictly increasing subsequence, followed the values of the lexicographically-earliest such sequence. A sequence $a_1, a_2, \ldots, a_m$ is lexicographically earlier than $b_1, b_2, \ldots, b_m$ if some $a_i < b_i$ and $a_j = b_j$ for all $j < i$.

Sample 1

Input	Output
4 1 25 2 3 4 1 2 2 3 8 90 4 10000 2 18 60 172 99 0	3 1 2 3 3 1 2 3 4 2 18 60 99