Problem F. Full Irreducibility

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

A permutation is called *irreducible* if none of its prefixes form a permutation, except the full permutation itself. For example, $[2, 3, 1]$ and $[4, 1, 2, 3]$ are irreducible, while $[2, 1, 3]$ and $[1, 3, 2]$ are not.

You are given a permutation p of length n. In one operation, you can choose any two adjacent positions and swap the values at these positions.

Find the minimum number of operations to transform p into an irreducible permutation, and the corresponding sequence of operations itself.

Input

The first line of input contains a single integer t, the number of test cases ($1 \le t \le 10^5$).

The first line of each test case contains a single integer n ($1 \le n \le 10^5$).

The second line of each test case contains n integers $p_1, \ldots, p_n$ ($1 \le p_i \le n$).

It is guaranteed that the sum of n over all test cases does not exceed $5 \cdot 10^5$.

Output

For each test case, print two lines:

On the first line, print an integer k: the minimum number of operations that make p irreducible.

On the second line, print k space-separated integers $s_1, \ldots, s_k$ ($1 \le s_i \le n - 1$) where s_i and $s_i + 1$ are the positions of the values you intend to swap. The operations are performed sequentially in the order specified by your output. If there are multiple solutions, print any one of them.

Example

standard input	*standard output*
3 4 1 2 3 4 3 2 3 1 5 3 1 2 4 5	3 1 2 3 0 2 4 3