
Heap Partition

Input file: **standard input**
Output file: **standard output**
Time limit: 3 seconds
Memory limit: 256 megabytes

A sequence $S = \{s_1, s_2, \dots, s_n\}$ is called *heapable* if there exists a binary tree T with n nodes such that every node is labelled with exactly one element from the sequence S , and for every non-root node s_i and its parent s_j , $s_j \leq s_i$ and $j < i$ hold. Each element in sequence S can be used to label a node in tree T only once.

Chiaki has a sequence $a_1, a_2, \dots, a_n$, she would like to decompose it into a minimum number of heapable subsequences.

Note that a subsequence is a sequence that can be derived from another sequence by deleting some elements without changing the order of the remaining elements.

Input

There are multiple test cases. The first line of input contains an integer T , indicating the number of test cases. For each test case:

The first line contain an integer n ($1 \leq n \leq 10^5$) – the length of the sequence.

The second line contains n integers $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq n$).

It is guaranteed that the sum of all n does not exceed 2×10^6 .

Output

For each test case, output an integer m denoting the minimum number of heapable subsequences in the first line. For the next m lines, first output an integer C_i , indicating the length of the subsequence. Then output C_i integers $P_{i1}, P_{i2}, \dots, P_{iC_i}$ in increasing order on the same line, where P_{ij} means the index of the j -th element of the i -th subsequence in the original sequence.

Example

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