

Problem G. Quicksort

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 512 mebibytes

You are given an array of numbers a_1, a_2, \dots, a_n printed on a set of cards that is arranged in a row, and need to sort it in non-decreasing order. The cards are pretty big, so one person can only swap two adjacent cards. However, you have as many people available to do the swapping as you need, as long as they don't try to swap the same card, so in one step you can do swaps on an arbitrary disjoint set of pairs of adjacent cards.

More formally, in one step you can choose a set of indices i_1, i_2, \dots, i_k such that $i_1 + 2 \leq i_2$, $i_2 + 2 \leq i_3$, \dots , $i_{k-1} + 2 \leq i_k$, and swap a_{i_1} with a_{i_1+1} , a_{i_2} with a_{i_2+1} , \dots , a_{i_k} with a_{i_k+1} .

Your goal is to sort the given array in non-decreasing order in at most n steps. Note that you don't need to minimize the number of steps.

Input

The input contains multiple test cases. The first line of the input contains the number t of test cases. Each test case is then described with two lines. The first of those lines contains the size n of the array, $1 \leq n \leq 100$. The second of those lines contains the elements of the array a_1, a_2, \dots, a_n . The elements of the array are integers between 1 and 10^9 , inclusive.

The sum of values of n in the input does not exceed 1000.

Output

Print the outputs for all test cases in order. For each test case, first print the number m ($0 \leq m \leq n$) of steps that you use to sort the array on a line of its own. On each of the next m lines describe one step with the number k ($0 \leq k \leq \lfloor \frac{n}{2} \rfloor$) of pairs being swapped followed by the indices i_1, i_2, \dots, i_k of the first elements of each pair, printed in increasing order.

Example

standard input	standard output
2	3
3	1 1
3 2 1	1 2
4	1 1
300 200 400 100	3
	2 1 3
	1 2
	1 1