Problem E. Oscar is All You Need

Input file:	standard input
Output file:	standard output
Time limit:	1 second
Memory limit:	1024 megabytes

Putata has a sequence p of length n, where p is a permutation of $1, 2, \ldots, n$. Budada can perform the following operation at most $2n + 1$ times:

- Cut the sequence into three consecutive non-empty parts, and swap the first part and the last part. Formally, you should select two integers x, y satisfying that $x > 0$, $y > 0$, $x + y < n$, and the sequence will change from $p_1, \ldots, p_x, p_{x+1}, \ldots, p_{n-y}, p_{n-y+1}, \ldots, p_n$ to $p_{n-y+1}, \ldots, p_n, p_{x+1}, \ldots, p_{n-y}, p_1, \ldots, p_x$.

Budada wants to make the lexicographical order of the permutation as small as possible after no more than $2n + 1$ operations. Please help him find the way to perform operations so that the lexicographical order of the permutation is as small as possible.

A **permutation** is an array where each integer from 1 to s (where s is the size of permutation) occurs exactly once.

A permutation a is lexicographically smaller than a permutation b if and only if the following condition holds:

- Let x be the smallest integer where $a_y = b_y$ holds for all $y \le x$, then we have $x < n$ and $a_{x+1} < b_{x+1}$.

Input

The input contains several test cases.

The first line contains an integer T ($1 \le T \le 120$), denoting the number of test cases.

For each test case, the first line contains an integer n ($3 \le n \le 1000$), denoting the length of the permutation.

The second line contains n integers, the i-th integer is p_i ($1 \le p_i \le n$), denoting the permutation. It is guaranteed that p is a permutation of $1, 2, \ldots, n$.

It is guaranteed that the sum of n in all test cases will not exceed 1000.

Output

For each test case, output one integer m in the first line, denoting the number of operations. You should guarantee that $0 \le m \le 2n + 1$.

Then output m lines, each line contains two integers x, y, denoting one operation. You should guarantee that $0 < x$, $0 < y$, $x + y < n$.

Please notice that you **do not have to** minimize the number of operations.

Example

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