Problem
Keep It Sorted
Time limit: 1 seconds

Gon has a permutation $a = (a_1, a_2, \ldots, a_n)$ of integers between 1 and n (inclusive), and wants to sort it in increasing order. However, sorting is trivial—anyone can sort an array! Thus, Gon decides to only use the following operation to sort:

- Select 2 indices ℓ and r ($1 \leq \ell \leq r \leq n$), such that the sub-array $(a_\ell, a_{\ell+1}, \ldots, a_r)$ is sorted, in either increasing or decreasing order.
- Reverse the sub-array from ℓ to r.

For example, given a permutation $a = (3, 2, 1)$, Gon can sort it with 1 operation as follow:

- Select $\ell = 1, r = 3$.
- Reverse the sub-array (a_1, a_2, a_3) to get $a = (1, 2, 3)$.

Note that if $a = (3, 1, 2)$, Gon cannot select $\ell = 1, r = 3$, as the sub-array from 1 to 3 is not sorted.

As Gon's birthday is Jan 19th, Gon wants to use at most 191 operations. This turns out to be non-trivial. Please help Gon!

Input

The first line of input contains a single integer n ($1 \leq n \leq 100$).

The second line of input contains n integer $a_1, a_2, \ldots, a_n$. It is guaranteed that a is a permutation of integers between 1 and n, inclusive.

Output

Print a single number k ($0 \leq k \leq 191$) on the first line—the number of operations you want to use.

In each of the next k lines, print two integers ℓ and r ($1 \leq \ell \leq r \leq n$) describing the operations. The sub-array between ℓ and r must be sorted before this operation.

After k operations, the permutation a must be sorted in increasing order. In other words, $a_i = i$ for every valid index i. Note that you do not need to minimize the number of operations.

It is guaranteed that a solution exists. If there are multiple solutions, you can print any of them.

Sample Input 1	Sample Output 1
3 3 1 2	7 1 1 2 2 3 3 1 2 2 3 1 3 1 3

This page is intentionally left blank.