

# Problem F. Inversion SwapSort

**Time limit** 2000 ms

**Mem limit** 262144 kB

Madeline has an array  $a$  of  $n$  integers. A pair  $(u, v)$  of integers forms an inversion in  $a$  if:

- $1 \leq u < v \leq n$ .
- $a_u > a_v$ .

Madeline recently found a magical paper, which allows her to write two indices  $u$  and  $v$  and swap the values  $a_u$  and  $a_v$ . Being bored, she decided to write a list of pairs  $(u_i, v_i)$  with the following conditions:

- all the pairs in the list are distinct and form an inversion in  $a$ .
- all the pairs that form an inversion in  $a$  are in the list.
- Starting from the given array, if you swap the values at indices  $u_1$  and  $v_1$ , then the values at indices  $u_2$  and  $v_2$  and so on, then after all pairs are processed, the array  $a$  will be sorted in **non-decreasing order**.

Construct such a list or determine that no such list exists. If there are multiple possible answers, you may find any of them.

## Input

The first line of the input contains a single integer  $n$  ( $1 \leq n \leq 1000$ ) — the length of the array.

Next line contains  $n$  integers  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^9$ ) — elements of the array.

## Output

Print  $-1$  if no such list exists. Otherwise in the first line you should print a single integer  $m$  ( $0 \leq m \leq \frac{n(n-1)}{2}$ ) — number of pairs in the list.

The  $i$ -th of the following  $m$  lines should contain two integers  $u_i, v_i$  ( $1 \leq u_i < v_i \leq n$ ).

If there are multiple possible answers, you may find any of them.

## Sample 1

Input	Output
3 3 1 2	2 1 3 1 2

**Sample 2**

Input	Output
4 1 8 1 6	2 2 4 2 3

**Sample 3**

Input	Output
5 1 1 1 2 2	0

**Note**

In the first sample test case the array will change in this order  $[3, 1, 2] \rightarrow [2, 1, 3] \rightarrow [1, 2, 3]$ .

In the second sample test case it will be  $[1, 8, 1, 6] \rightarrow [1, 6, 1, 8] \rightarrow [1, 1, 6, 8]$ .

In the third sample test case the array is already sorted.