

## B. Little Robber Girl's Zoo

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Little Robber Girl likes to scare animals in her zoo for fun. She decided to arrange the animals in a row in the order of non-decreasing height. However, the animals were so scared that they couldn't stay in the right places.

The robber girl was angry at first, but then she decided to arrange the animals herself. She repeatedly names numbers  $l$  and  $r$  such that  $r - l + 1$  is even. After that animals that occupy positions between  $l$  and  $r$  inclusively are rearranged as follows: the animal at position  $l$  swaps places with the animal at position  $l + 1$ , the animal  $l + 2$  swaps with the animal  $l + 3$ , ..., finally, the animal at position  $r - 1$  swaps with the animal  $r$ .

Help the robber girl to arrange the animals in the order of non-decreasing height. You should name at most 20 000 segments, since otherwise the robber girl will become bored and will start scaring the animals again.

### Input

The first line contains a single integer  $n$  ( $1 \leq n \leq 100$ ) — number of animals in the robber girl's zoo.

The second line contains  $n$  space-separated integers  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^9$ ), where  $a_i$  is the height of the animal occupying the  $i$ -th place.

### Output

Print the sequence of operations that will rearrange the animals by non-decreasing height.

The output should contain several lines,  $i$ -th of the lines should contain two space-separated integers  $l_i$  and  $r_i$  ( $1 \leq l_i < r_i \leq n$ ) — descriptions of segments the robber girl should name. The segments should be described in the order the operations are performed.

The number of operations should not exceed 20 000.

If the animals are arranged correctly from the start, you are allowed to output nothing.

### Examples

<b>input</b>
4 2 1 4 3
<b>output</b>
1 4
<b>input</b>
7 36 28 57 39 66 69 68
<b>output</b>
1 4 6 7
<b>input</b>
5 1 2 1 2 1
<b>output</b>
2 5 3 4

### Codeforces Round #359 (Div. 2)

Finished

#### → Virtual participation

Virtual contest is a way to take part in past contest, as close as possible to participation on time. It is supported only ACM-ICPC mode for virtual contests. If you've seen these problems, a virtual contest is not for you - solve these problems in the archive. If you just want to solve some problem from a contest, a virtual contest is not for you - solve this problem in the archive. Never use someone else's code, read the tutorials or communicate with other person during a virtual contest.

Start virtual contest

#### → Problem tags

constructive algorithms

implementation

sortings

No tag edit access

#### → Contest materials

• Announcement



• Tutorial



14  
14

### Note

Note that you don't have to minimize the number of operations. Any solution that performs at most 20 000 operations is allowed.

---

[Codeforces](#) (c) Copyright 2010-2016 Mike Mirzayanov  
The only programming contests Web 2.0 platform  
Server time: Nov/30/2016 19:22:26<sup>UTC+8</sup> (c4).  
Desktop version, switch to [mobile version](#).