D. Grocer's Problem

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Yesterday was a fair in a supermarket's grocery section. There were n jars with spices on the fair. Before the event the jars were numbered from 1 to n from the left to the right. After the event the jars were moved and the grocer had to sort them by the increasing of the numbers.

The grocer has a special machine at his disposal. The machine can take any 5 or less jars and rearrange them in the way the grocer wants. Note that the jars **do not have to** stand consecutively. For example, from the permutation $2, 6, 5, 4, 3, 1$ one can get permutation $1, 2, 3, 4, 5, 6$, if pick the jars on the positions $1, 2, 3, 5$ and 6.

Which minimum number of such operations is needed to arrange all the jars in the order of their numbers' increasing?

Input

The first line contains an integer n $(1 \le n \le 10^5)$. The second line contains n space-separated integers a_i $(1 \le a_i \le n)$ — the i-th number represents the number of a jar that occupies the i-th position. It is guaranteed that all the numbers are distinct.

Output

Print on the first line the least number of operations needed to rearrange all the jars in the order of the numbers' increasing. Then print the description of all actions in the following format.

On the first line of the description of one action indicate the number of jars that need to be taken (k), on the second line indicate from which positions the jars need to be taken ($b_1, b_2, ..., b_k$), on the third line indicate the jar's new order ($c_1, c_2, ..., c_k$). After the operation is fulfilled the jar from position b_i will occupy the position c_i. The set ($c_1, c_2, ..., c_k$) should be the rearrangement of the set ($b_1, b_2, ..., b_k$).

If there are multiple solutions, output any.

Examples

input

```
6
3 5 6 1 2 4
```

output

```
2
4
1 3 6 4
3 6 4 1
2
2 5
5 2
```

input

```
14
9 13 11 3 10 7 12 14 1 5 4 6 8 2
```

output

```
3
4
2 13 8 14
13 8 14 2
5
6 7 12 5 10
7 12 6 10 5
```

```
5
3 11 4 1 9
11 4 3 9 1
```

Note

Let's consider the first sample. The jars can be sorted within two actions.

During the first action we take the jars from positions $1, 3, 6$ and 4 and put them so that the jar that used to occupy the position 1 will occupy the position 3 after the operation is completed. The jar from position 3 will end up in position 6, the jar from position 6 will end up in position 4 and the jar from position 4 will end up in position 1.

After the first action the order will look like that: $1, 5, 3, 4, 2, 6$.

During the second operation the jars in positions 2 and 5 will change places.