Task Volontiranje

It is not so well known that in his free time, Mr. Malnar cotributes to the community by volunteering. That's right! He is quite active at the volunteering center for informatics problems.

Recently, he got a call from the center and, as it turns out, there is a shortage of increasing sequences. Mr. Malnar, who is always willing to help, responded readily. Namely, he keeps an array of n integers for this very purpose. All of the integers from 1 to n appear exactly once in his array.

Mr. Malnar will select a few increasing subsequences from his array, which he will donate to the center, and he will keep the rest of the numbers for some other time. It must be noted that each number from the array can be used for at most one subsequence, that is, the selected subsequences have distinct indices.

A subsequence is defined as any sequence obtained from the array by deleting some (maybe none) of the elements, while preserving the order of the remaining elements. A subsequence is said to be increasing if every number in the subsequence (except for the first one) is larger than the previous one.

Because Mr. Malnar is very generous, he wants all of the sequences he donates to be longest increasing subsequences. In other words, if l is the length of the longest increasing subsequence of the staring array, Mr. Malnar will select a couple of disjoint increasing subsequences, each of length l.

Mr. Malnar wants to donate as many subsequences as possible. For a given array of Mr. Malnar, print the maximum number of subsequences which comprise his selection and provide an example of such a selection.

Input

The first line contains the integer n - the size of the array.

The second line contains n numbers p_i $(1 \leq p_i \leq n)$ which represent the elements of the array. Each positive integer j between 1 and n appears exactly once in the array.

Output

In the first line, print the largest possible number of subsequences in Mr. Malnar's selection, as well as the length of the longest increasing subsequence.

In each of the following lines print one of the subsequences from such a selection. A subsequence is defined by a list of position, i.e. **indices** of the array, which should be **sorted in increasing order**.

The printed subsequences should be **increasing**, **disjoint** and have the **same length** - the length of the longest increasing subsequence.

The relative order of the subsequences does not matter. Also, if there is more than one solution, you can print any.

Scoring

In every subtask, it holds that $1 \leq n \leq 1\ 000\ 000$.

Subtask	Points	Constraints
1	10	$1 \le n \le 15$
2	40	$1 \le n \le 1000$
4	60	$1 \le n \le 1\ 000\ 000$

Examples

input

```
3
1 2 3
```

output

```
1 3
1 2 3
```

input

```
4
4 3 2 1
```

output

```
4 1
1
2
3
4
```

input

```
7
2 1 6 5 7 3 4
```

output

```
2 3
1 3 5
2 6 7
```

Clarification of the third example:
The length of the longest increasing subsequence is 3. The subsequence determined by indices 1, 3, and 5 (or, values 2, 6 and 7) is increasing. The subsequence determined by indices 2, 6 and 7 (or, values 1, 3 and 4) is also incrasing. The two subsequences have no elements in common and they also have maximum lenght, which is why this is a valid selection. It is not possible to choose more than two of such subsequences.