Task F. Cycle sort

You are given an array of n positive integers $a_1, a_2, \ldots, a_n$. You can perform the following operation any number of times: select several distinct indices $i_1, i_2, \ldots, i_k$ ($1 \le i_j \le n$) and move the number standing at the position i_1 to the position i_2, the number at the position i_2 to the position i_3, $\ldots$, the number at the position i_k to the position i_1. In other words, the operation cyclically shifts elements: $i_1 \to i_2 \to \ldots i_k \to i_1$.

For example, if you have $n = 4$, an array $a_1 = 10, a_2 = 20, a_3 = 30, a_4 = 40$, and you choose three indices $i_1 = 2$, $i_2 = 1$, $i_3 = 4$, then the resulting array would become $a_1 = 20, a_2 = 40, a_3 = 30, a_4 = 10$.

Your goal is to make the array sorted in non-decreasing order with the minimum number of operations. The additional constraint is that the sum of cycle lengths over all operations should be less than or equal to a number s. If it's impossible to sort the array while satisfying that constraint, your solution should report that as well.

Input

The first line of the input contains two integers n and s ($1 \le n \le 200\,000$, $0 \le s \le 200\,000$), the number of elements in the array and the upper bound on the sum of cycle lengths.

The next line contains n integers $a_1, a_2, \ldots, a_n$, the elements of the array ($1 \le a_i \le 10^9$).

Output

If it's impossible to sort the array using cycles of total length not exceeding s, print a single number `-1`.

Otherwise, print a single number q, the minimum number of operations required to sort the array. On the next $2 \cdot q$ lines print the description of the operations in the order they are applied to the array. The description of i-th operation begins with a single line containing one integer k ($1 \le k \le n$)—the length of the cycle (that is, the number of selected indices). The next line should contain k distinct integers $i_1, i_2, \ldots, i_k$ ($1 \le i_j \le n$)—the indices themselves.

The sum of lengths of these cycles should be less than or equal to s, and the array should be sorted after applying these q operations.

If there are several possible answers with the optimal q, print any of them.

Scoring

This problem contains nine subtasks, for each subtask you will get the points only if you pass all the tests for this subtask.

1. (5 points) $n, s \leq 2$ and all elements of the array are either 1 or 2
2. (5 points) $n \leq 5$
3. (5 points) All elements of the array are either 1 or 2
4. (10 points) The array contains numbers from 1 to n only, each number appears exactly once, $s = 2 \cdot n$
5. (10 points) The array contains numbers from 1 to n only, each number appears exactly once, $n \leq 1000$
6. (15 points) The array contains numbers from 1 to n only, each number appears exactly once
7. (15 points) $s = 2 \cdot n$
8. (15 points) $n \leq 1000$
9. (20 points) No additional constraints.

Examples

Example 1

Input:

```
5 5
3 2 3 1 1
```

Output:

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

In this example, it's also possible to sort the array with two operations of total length 5: first apply the cycle $1 \to 4 \to 1$ (of length 2), then apply the cycle $2 \to 3 \to 5 \to 2$ (of length 3). However, it would be a wrong answer as you're asked to use the minimal possible number of operations, which is 1 in that case.

Example 2

Input:

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

Output:

```
-1
```

In this example, it's possible to the sort the array with two cycles of total length 4 ($1 \to 2 \to 1$ and $3 \to 4 \to 3$). However, it's impossible to achieve the same using shorter cycles, which is required by $s = 3$.

Example 3

Input:

```
2 0
2 2
```

Output:

```
0
```

In this example, the array is already sorted, so no operations are needed. The total length of empty set of cycles is considered to be zero.

Example 4

Input:

```
6 9
6 5 4 3 2 1
```

Output:

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

Example 5

Input:

```
6 8
6 5 4 3 2 1
```

Output:

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

Notice that examples 1 and 3 contain duplicate numbers, so they do not satisfy the requirements for subtasks 4, 5 and 6. Examples 2, 4, and 5 satisfy the requirements for subtasks 5 and 6.