mmmm, Pancakes

While we've considered stacks and queues in the course, in this assessment you'll be dealing with a slightly different type of stack. More precisely, you're given a stack of pancakes, and your task is to write a program to sort this stack.

The difficulty is that your pancake sorting program has the use of only a single tool, a rather large spatula. All it can do is slide this spatula under a pancake at some point in the stack, and flip this pancake, and everything above it over.

For example, consider the following stack of pancakes:

```
4
3
8
5
2
```

Sliding the spatula under pancake 8 and flipping results in the following stack:

```
8
3
4
5
2
```

We will call the bottom most position in the stack position 1, and the topmost point position n (where n is the number of pancakes).

Now, given a set of pancakes of different sizes (as signified by the numbers above), your task is to provide a sequence of flip operations that will result in the largest pancake on the bottom, the second largest above that, and so on, until the smallest pancake rests on the top. For example, the stack above would be considered sorted when it lies in the following order:

```
2
3
4
5
8
```

Input

The input consists of a sequence of stacks of pancakes, with each stack on a new line. The input is terminated by a single 0 on its own line. The bottom pancake appears as the first element in the list, the second as the second element, and so on, each separated by a space.

Output

For each pancake stack, the output should first print the original stack on a single line, followed by another line identifying the sequence of flips that results in the stack of pancakes being sorted so that the largest pancake is on the bottom. This sequence of flips should be terminated by a 0. If a stack is sorted, no additional flips should be made.

A flip gives the index of the pancake under which the spatula should be placed, with the bottom pancake having index 1. In other words, providing a 1 in the output means that the whole stack will be flipped.

When no more of the sequence remains, you should output a 0 on its own

Sample Input

```
5 4 3 2 1
1 2 3 4 5
4 3 2 1 5
0
```

Sample Output

```
5 4 3 2 1
0
1 2 3 4 5
1 0
4 3 2 1 5
1 2 0
0
```