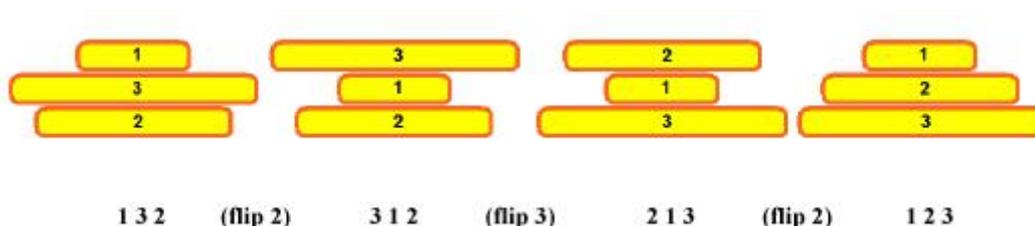


## 3145 Flipping Pancakes

We start with a stack  $n$  of pancakes of distinct sizes. The problem is to convert the stack to one in which the pancakes are in size order with the smallest on the top and the largest on the bottom. To do this, we are allowed to flip the top  $k$  pancakes over as a unit (so the  $k$ -th pancake is now on top and the pancake previously on top is now in the  $k$ -th position).

For example:



This problem is to write a program, which finds a sequence of at most  $(2n - 3)$  flips, which converts a given stack of pancakes to a sorted stack.

### Input

Each line of the input gives a separate data set as a sequence of numbers separated by spaces. The first number on each line gives the number,  $N$ , of pancakes in the data set. The input ends when  $N$  is 0 (zero) with no other data on the line. The remainder of the data set are the numbers 1 through  $N$  in some order giving the initial pancake stack.

The numbers indicate the relative sizes of the pancakes.  $N$  will be, at most, 30.

### Output

For each data set, the output is a single-space separated sequence of numbers on a line. The first number on each line,  $K$ , gives the number of flips required to sort the pancakes. This number is followed by a sequence of  $K$  numbers, each of which gives the number of pancakes to flip on the corresponding sorting step.

There may be several correct solutions for some datasets. For instance '3 3 2 3' is also a solution to the first problem below.

### Sample Input

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

### Sample Output

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