From Skiena and Revilla, PROGRAMMING CHALLENGES: *The Programming Contest Training Manual*
Springer-Verlag, New York, 2003. ISBN: 0-387-00163-8

110402 Stacks of Flapjacks

Cooking the perfect stack of pancakes on a grill is a tricky business, because no matter how hard you try all pancakes in any stack have different diameters. For neatness's sake, however, you can sort the stack by size such that each pancake is smaller than all the pancakes below it. The size of a pancake is given by its diameter.

Sorting a stack is done by a sequence of pancake "flips." A flip consists of inserting a spatula between two pancakes in a stack and flipping (reversing) *all* the pancakes on the spatula (reversing the sub-stack). A flip is specified by giving the position of the pancake on the bottom of the sub-stack to be flipped relative to the entire stack. The bottom pancake has position 1, while the top pancake on a stack of n pancakes has position n.

A stack is specified by giving the diameter of each pancake in the stack in the order in which the pancakes appear. For example, consider the three stacks of pancakes below in which pancake 8 is the top-most pancake of the left stack:

```
8          7          2
4          6          5
6          4          8
7          8          4
5          5          6
2          2          7
```

The stack on the left can be transformed to the stack in the middle via *flip(3)*. The middle stack can be transformed into the right stack via the command *flip(1)*.

Input

The input consists of a sequence of stacks of pancakes. Each stack will consist of between 1 and 30 pancakes and each pancake will have an integer diameter between 1 and 100. The input is terminated by end-of-file. Each stack is given as a single line of input with the top pancake on a stack appearing first on a line, the bottom pancake appearing last, and all pancakes separated by a space.

Output

For each stack of pancakes, your program should echo the original stack on one line, followed by a sequence of flips that results in sorting the stack of pancakes so that the largest pancake is on the bottom and the smallest on top. The sequence of flips for each stack should be terminated by a 0, indicating no more flips necessary. Once a stack is sorted, no more flips should be made.

Sample Input

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

Sample Output

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