Transform a Sequence

An integer sequence is called **block** iff it is made of two identical and neighboring sequences. For example, (3, 3) and (5, 2, 5, 2) are blocks, while (9, 6, 6, 9) and (1, 1, 1) are not blocks.

An integer sequence is called **non-descreasing** iff its each element (except the last one) is less than or equal to the next one. A sequence containing one or zero elements is also non-decreasing.

You are given an integer sequence and you are to transform it into a **non-decreasing** sequence by multiple using of two types of operations:

Type 1: removing a block from the sequence (if that block is a consecutive subsequence);

Type 2: inserting a block into the sequence (so that it becomes a consecutive subsequence).

In all of the test data there will be at least one solution.

Input

In the first line there is an integer N ($3 \leq N \leq 40$), the size of the given sequence.

In the next line there are N space-separated integers (of size 0 - 100), the elements of the sequence (in order).

Output

Output at most 2340 operations.

For each operation, write two or three lines (depending on the type of operation). In the first line write the type of the operation (1 or 2).

If the type of the operation is 1 (removing the block), in the next line write the two numbers: positions in the sequence of the first and the last element of the block you are removing.

If the type of the operation is 2 (adding the block), in the second line write the two numbers: positions in the sequence of the first and the last element of the block after it is added to the sequence. In the next line, write all the elements of the added block (in order). The elements must be integers from the interval [-1000, 1000].

There must be less than 1000 elements in the sequence at any time.

Example

Input:

6

10 20 70 20 70 80

Output:

1

2 5

Input:

6

0 4 0 4 4 0

Output:

2

3 4

4 4

1

3 8