Task: TET
Tetris Attack

Stage II. Day 2. Source file `tet.*` *08.02.2007*

Available memory: 32 MB.

A puzzle called “Tetris Attack” has lately become a very popular game in Byteotia. The game itself is highly sophisticated, so we shall only introduce its simplified rules: the player is given a stack of $2n$ elements, placed one on another and marked with n different symbols. Exactly two elements of the stack are marked with each symbol. A player’s move consists in swapping two neighbouring elements, ie. interchanging them. If, as a result of the swap, there are some two neighbouring elements marked with the same symbol, they are both removed from the stack. Then all the elements that have been above them fall down in consequence and may very well cause another removals. The player’s goal is emptying the stack in the least possible number of moves.

Task

Write a programme that:

- reads the description of the initial stack content from the standard input,
- finds a solution with the minimum number of moves possible,
- writes out the outcome to the standard output.

Input

In the first line of the standard input there is one integer n, $1 \leq n \leq 50\,000$. The following $2n$ lines describe the initial content of the stack. The $(i+1)^{th}$ line contains one integer a_i — the symbol which the i^{th} $(1 \leq a_i \leq n)$ element from the bottom is marked with. Each symbol appears in the stack **exactly** twice. Moreover, no two identical symbols neighbour initially. The test data is well chosen so that a solution with no more than $1\,000\,000$ moves exists.

Output

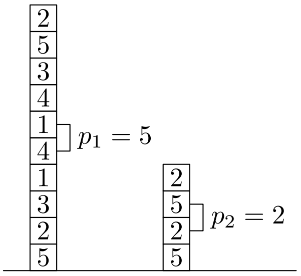
A solution with the minimum number of moves possible should be written out to the standard output as follows. The first line should contain one integer m — the minimum number of moves. The following m should describe the optimal solution itself, i.e. a sequence of m integers $p_1, \ldots, p_m$, one in each line. p_i denotes that in i^{th} move the player has chosen to swap the p_i^{th} and $(p_i+1)^{th}$ elements from the bottom.

If more than one optimal solution exists, your programme should write out any one of them.

Example

For the input data:

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



the correct result is:

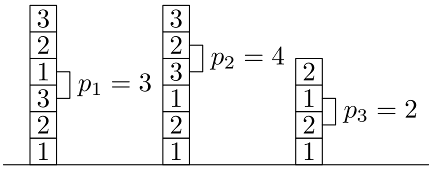
```
2
5
2
```

While for the following input data:

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

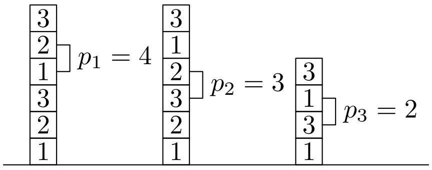
the correct result is:

```
3
3
4
2
```



as well as:

```
3
4
3
2
```



3
2
1
3
2
1
$p_1 = 4$
3
1
2
3
2
1
$p_2 = 3$
3
1
3
1
$p_3 = 2$