Sort by Adjacent Swaps

You are given an array A

(of size n) and you can only perform the adjacent swaps on the array, i.e. you can choose an index i ($0 \le i \le n-2$) and swap the values present in Ai and $Ai+1$

.

Sort the array using **minimum** number of adjacent swap operations.

Note that there may be multiple possible outputs, you can print **any** of them.

Input

The first line contains a single integer n

, the number of elements in the array

The second line contains n

space separated integers, the elements of the array

Output

In the first line print the number of swaps that your solution performs on the array, let this number be m

In the next line print m

space separated integers, each between 0 and $n-2$ (inclusive), the ith integer denotes that Ai and $Ai+1$

are swapped.

Note that m

must be equal to the minimum number of adjacent swap operations required to sort the array.

Constraints

$1 \le n \le 105$

$-109 \le Ai \le 109$

It is guaranteed that for the given array, the minimum number of swaps required is no more than 105

Sample Input

```
3
3 2 1
```

Sample Output

```
3
0 1 0
```

Explanation

The first operation swaps the first two elements of the array ($A0$

and $A1$), so the array becomes $\{2,3,1\}$

The second operation swaps the second and third elements of the array, so the array becomes $\{2,1,3\}$

The third operation swaps the first two elements of the array, so the array becomes $\{1,2,3\}$

You can verify that it cannot be done in less than 3

operations.