Swap Sort

You are given an array A

of size n (not exceeding 500), you are allowed a maximum of $n-1$ swaps where in each swap you can choose **any two** indices i,j such that $0 \le i,j \le n-1$ and swap the values of Ai and Aj

.

Note that you **do not** need to minimise the number of swaps, and you are also **not required** to optimize the time complexity (note that the constraints allow solutions that are $O(n2)$

as well), however you **cannot** swap more than $n-1$

times.

There may be many solutions, you can output **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 you need to perform, let this number be m

.

In the next m

lines, print two space separated integers i,j such that $0 \le i,j \le n-1$

.

After performing the swaps in the order in which your program gives the output, the array should become sorted in increasing order. (i.e. $Ai \ge Ai-1$

for every i in the range $[1,n-1]$

).

Constraints

$1 \le n \le 500$

$-109 \le Ai \le 109$

Sample Input

```
3
3 2 1
```

Sample Output

```
1
0 2
```

Explanation

After the first swap, the elements $A0$

and $A2$ are swapped, and the array becomes sorted, i.e. $\{1,2,3\}$