

Inversion Counting (Small)

CITS3001 Advanced Algorithms — Time limit: 5 s

An *inversion* in a list is a pair of positions $i < j$ whose elements are out of order: $a_i > a_j$. The number of inversions measures how far a list is from being sorted: a sorted list has none, and a reverse-sorted list has as many as possible.

Given a list, count its inversions. For this problem the list is small, so an $O(N^2)$ algorithm — for example, your modified insertion sort from earlier in the lab — is fast enough.

Input

The first line contains an integer N ($1 \leq N \leq 1000$). The second line contains N integers $a_1, \dots, a_N$ ($0 \leq a_i \leq 10^9$).

Output

Output a single integer: the number of pairs $i < j$ with $a_i > a_j$. Note that equal elements do *not* form an inversion.

Sample Input 1

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

Sample Output 1

```
3
```

Sample Input 2

```
3
3 3 3
```

Sample Output 2

```
0
```