Inversion Counting

Input File	Output File	Time Limit	Memory Limit
standard input	standard output	1 second	256 MiB

Everyone likes sorted arrays, but sometimes they are not sorted :(

However, some unsorted arrays are closer to being sorted than others! Let's define a formal measure for how close to sorted an array is.

Suppose we have an array of N elements. The i-th element is A_i (indexed starting from 1). A pair of indices i and j ($i < j$) are said to be *inverted* if $A_i > A_j$.

For example, in the array `[1, 9, 1, 7]`, the pair of indices 2 and 4 are inverted (since $9 > 7$).

The *inversion count* of an array is the total number of pairs of indices that are inverted. Can you calculate the inversion count?

Input

- The first line of input contains the integer N ($1 \leq N \leq 10^5$).
- The second line contains N integers, describing the array. The i-th integer is A_i ($1 \leq A_i \leq 10^5$).

Output

Output the total number of inversions. This number might be quite large, so you might want to use a `long long int` (or the equivalent for your language).

Sample Input 1

```
4
1 9 1 7
```

Sample Output 1

```
2
```

Sample Input 2

```
6
1 1 2 2 3 3
```

Sample Output 2

```
0
```

Sample Input 3

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

Sample Output 3

```
10? Something like that.
```