Problem H. How many inversions?

Input:	`standard`
Output:	`standard`
Author(s):	`Hugo Humberto Morales Peña - UTP Colombia`

Humbertov Moralov in his student days, he is attended system engineering at "University of missing hill". He was evaluated in its first course of Analysis of Algorithms (at the first half of 1997) with the following topics and questions:

Inversions :

Let $A[1 \dots n]$ an array of distinct integers of size n. If $i < j$ and $A[i] > A[j]$, then the pair (i, j) is called an **inversion** of A.

Given the above definition about an inversion, *Humbertov Moralov* must answer the following questions:

1. List all inversions in $\langle 3, 2, 8, 1, 6 \rangle$.
2. What array of size n, with all the numbers from the set $1, 2, 3, \dots, n$ has the largest amount of inversions? How many inversions?
3. Write an algorithm to determine the number of inversions in any permutation of n elements with $\theta(n \log n)$ in the worst case run time.

Humbertov Moralov answered questions 1. and 2. without any problem, but he was not able to solve the question 3 at time. Days later he thought the following solution:

```
1:  inv ← 0
2:  function MERGE(A[], p, q, r)
3:      n1 ← q − p + 1
4:      n2 ← r − q
5:      create arrays L[1 ... n1 + 1] and R[1 ... n2 + 1]
6:      for i = 1 to n1 do
7:          L[i] ← A[p + i − 1]
8:      end for
9:      for j = 1 to n2 do
10:         R[j] ← A[q + j]
11:     end for
12:     L[n1 + 1] ← ∞
13:     R[n2 + 1] ← ∞
14:     i ← 1
15:     j ← 1
16:     for k = p to r do
17:         if L[i] ≤ R[j] then
18:             A[k] ← L[i]
19:             i ← i + 1
20:         else
21:             A[k] ← R[j]
22:             j ← j + 1
```

```
23:             inv ← inv + n1 − i + 1
24:         end if
25:     end for
26: end function
27: function MergeSort(A[], p, r)
28:     if p < r then
29:         q ← ⌊(p + r)/2⌋
30:         MergeSort(A[], p, q)
31:         MergeSort(A[], q + 1, r)
32:         Merge(A[], p, q, r)
33:     end if
34: end function
```

Will this code solve the problem? Just adding the lines 1 and 23 will be enough to solve the problem?

Please help *Humbertov Moralov* to validate this solution! For this, you must implement this solution in any of the programming languages accepted by the ACM-ICPC and verify if the expected results are generated.

Input

The input may contain several test cases. Each input case begins with a positive integer n ($1 \leq n \leq 10^6$) denoting the length of A, followed by n distinct lines. Each line contains a positive integer from array A. For $i \in [1, n]$, will meet that $0 \leq A[i] \leq 10^8$. The input ends with a test case in which n is zero, and this case must not be processed.

Output

For each test case, your program must print a positive integer representing the total number of inversions in the array A. Each valid test case must generate just one output line.

Example

Input	Output
5 3 2 8 1 6 5 5 4 3 2 1 1 10 0	5 10 0

Use faster I/O methods