



Allenamento su ChinaForces (allenamento)

As everyone knows, Filippo is training on the secret competitive programming website *ChinaForces* to improve quickly. Filippo has taken part in N contests: in the i -th of them, he achieved a performance A_i . The performances are distinct.



Figure 1: The *ChinaForces* logo

Filippo wants to analyze his performances in past contests to measure his improvement. Therefore, Filippo decides that a subarray of contests A_l, A_{l+1}, \dots, A_r is **improvementous** if and only the following condition is true:

- There exist two indices x, y ($l \leq x < y \leq r$) such that $A_x = \min(A_l, A_{l+1}, \dots, A_r)$ and $A_y = \max(A_l, A_{l+1}, \dots, A_r)$.

Help Filippo count the improvementous subarrays.

Implementation

You should submit a single file, with a `.cpp` extension.

📎 Among the attachments in this task you will find a template `allenamento.cpp` with a sample implementation.

You will have to implement the following function:

```
C++ | long long conta(int N, vector<int> A);
```

- The integer N represents the number of contests.
- The vector A , indexed from 0 to $N - 1$, contains the performance in each contest. In particular, for each $0 \leq i < N$, A_i is the performance in the i -th contest.

The function `conta` must return the number of improvementous subarrays of A .

Sample Grader

Among this task's attachments you will find a simplified version of the grader used during the evaluation, which you can use to test your solutions locally. The sample grader reads data from `stdin`, calls the functions that you should implement and writes back on `stdout` using the following format.

The input file consists of 2 lines, containing:

- Line 1: the integer N .
- Line 2: the N integers A_0, \dots, A_{N-1} .

The output file consists of a single line, containing the value returned by `conta`.

Constraints

- $1 \leq N \leq 10^7$.
- $1 \leq A_i \leq 10^9$.
- The A_i are distinct.

Scoring

Your program will be tested against several test cases grouped in subtasks. In order to obtain the score of a subtask, your program needs to correctly solve all of its test cases.

- **Subtask 1** [0 points]: Examples.
- **Subtask 2** [12 points]: $N \leq 20$
- **Subtask 3** [17 points]: $N \leq 400$
- **Subtask 4** [12 points]: $N \leq 1500$
- **Subtask 5** [17 points]: $N \leq 5000$
- **Subtask 6** [40 points]: $N \leq 2 \cdot 10^5$
- **Subtask 7** [2 points]: No additional limitations.

Examples

stdin	stdout
6 5 2 1 4 7 3	9
8 3 27 86 95 419 852 1473 6461	28

Explanations

In the **first example**, the improvementous subarrays are $[0, 4]$, $[0, 5]$, $[1, 3]$, $[1, 4]$, $[1, 5]$, $[2, 3]$, $[2, 4]$, $[2, 5]$, $[3, 4]$. For example:

- $[3, 4]$ is improvementous because $\min(A_3, A_4) = A_3$, $\max(A_3, A_4) = A_4$, therefore $x = 3$, $y = 4$: $x < y$ holds.
- $[1, 4]$ is improvementous because $\min(A_1, A_2, A_3, A_4) = A_2$, $\max(A_1, A_2, A_3, A_4) = A_4$, therefore $x = 2$, $y = 4$: $x < y$ holds.

In the **second example** all the subarrays of length at least 2 are improvementous.