**CHALLENGES** 

PRACTICE

**COMPANIES** 

All Tracks > Data Structures > Stacks > Basics of Stacks > Problem



First off, some definitions.

An array of length at least 2 having distinct integers is said to be fantabulous iff the second highest element lies **strictly to the left** of the highest value. For example, [1, 2, 13, 10, 15] is fantabulous as the second-highest value 13 lies to the left of highest value 15.

For every fantabulous array, we define a fantabulous pair (a, b) where a denotes the index of the second-highest value (1-indexed) and b denotes the index of the highest value (1-indexed). In the above array, the fantabulous pair is (3, 5).

Mancunian challenges you to solve the following problem. Given an array, find the total number of **distinct** fantabulous pairs over all its subarrays.

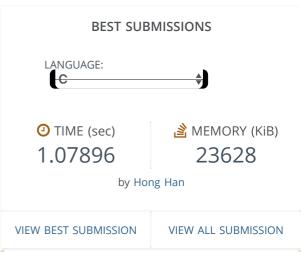
# Input:

The first line contains an integer **N** denoting the length of the array. The next line contains **N** distinct integers denoting the elements of the array.

## **Output:**

Output a single integer which is the answer to the problem.

## **Constraints:**



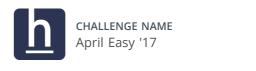
#### **CONTRIBUTOR**



Satyaki Upadhyay



# THIS PROBLEM WAS ASKED IN



## **SOCIAL SHARE**



$$1 \le N \le 10^6$$

1 <= array elements <=  $10^9$ 

Array elements are distinct.

SAMPLE INPUT	SAMPLE OUTPUT
4 1 3 2 4	3

# **Explanation**

Let us consider all the subarrays of the given array.

The subarray [1] is not fantabulous.

The subarray [2] is not fantabulous.

The subarray [3] is not fantabulous.

The subarray [4] is not fantabulous.

The fantabulous pair for subarray [1, 3] is (1, 2).

The subarray [3, 2] is not fantabulous.

The fantabulous pair for subarray [2, 4] is (1, 2).

The subarray [1, 3, 2] is not fantabulous.

The fantabulous pair for subarray [3, 2, 4] is (1, 3).

The fantabulous pair for subarray [1, 3, 2, 4] is (2, 4).

So, there are the 3 distinct pairs, which are (1, 2), (1,

3) and (2, 4).

Time Limit:	2.0 sec(s) for each input file.
Memory Limit:	256 MB
Source Limit:	1024 KB
Marking Scheme:	Marks are awarded when all the
	testcases pass.
Allowed Languages:	C, C++, C++14, Clojure, C#, D, Erlang,
	F#, Go, Groovy, Haskell, Java, Java 8,
	JavaScript(Rhino), JavaScript(Node.js),
	Julia, Kotlin, Lisp, Lisp (SBCL), Lua,
	Objective-C, OCaml, Octave, Pascal,
	Perl, PHP, Python, Python 3, R(RScript),
	Racket, Ruby, Rust, Scala, Swift, Visual
	Basic