**Java Subarray**

<https://www.hackerrank.com/challenges/java-negative-subarray/problem>

We define the following:

* A *subarray* of an *n*-element array is an array composed from a contiguous block of the original array's elements. For example, if *array = [1,2,3]*, then the subarrays are *[1]*, *[2]*, *[3]*, *[1,2]*, *[2,3]*, and *[1,2,3]*. Something like *[1,3]* would *not* be a subarray as it's not a contiguous subsection of the original array.
* The *sum* of an array is the total sum of its elements.
* An array's sum is *negative* if the total sum of its elements is negative.
* An array's sum is *positive* if the total sum of its elements is positive.

Given an array of *n* integers, find and print its number of *negative subarrays* on a new line.

**Input Format**

The first line contains a single integer, *n*, denoting the length of array *A = [a0, a1, . . . , an-1]*.  
The second line contains *n* space-separated integers describing each respective element, *ai*, in array *A*.

**Constraints**

* *1 <= n <= 100*
* *-104 <= ai <= 104*

**Output Format**

Print the number of subarrays of *A* having negative sums.

**Sample Input**

5

1 -2 4 -5 1

**Sample Output**

9

**Explanation**

There are nine negative subarrays of *A = [1,-2,4,-5,1]*:

1. *[1 : 1] => -2*
2. *[3 : 3] => -5*
3. *[0 : 1] => 1 + -2 = -1*
4. *[2 : 3] => 4 + -5 = -1*
5. *[3 : 4] => -5 + 1 = -4*
6. *[1 : 3] => -2 + 4 + -5 = -3*
7. *[0 : 3] => 1 + -2 + 4 + -5 = -2*
8. *[1 : 4] => -2 + 4 + -5 + 1 = -2*
9. *[0 : 4] => 1 + -2 + 4 + -5 + 1 = -1*

Thus, we print *9* on a new line.