Counting Perfect Subarrays

Problem code : CPFT
Time Limit : 1 second
Memory Limit : 256 MB

Problem Statement: An array is called **perfect** if no prefix exist in the array such that it's sum is negative . To understand perfection, let's look at all the prefix sums in the following arrays :

- $ar = [1, 2, -2, 3, -3]$, $prefix - sum = [1, 3, 1, 4, 1]$, as all the elements of prefix sum array is non-negative , ar is a **perfect** array.
- $ar = [10, -10]$, $prefix - sum = [10, 0]$, as all the elements of prefix sum array is non-negative , ar is a **perfect** array.
- $ar = [1, 2, -2, 3, -5, 3]$, $prefix - sum = [1, 3, 1, 4, -1, 2]$, as the $5th$ element of prefix sum array is negative , ar is **not** a perfect array.

Now given an array, you are required to count the number of sub-arrays present in the array that are perfect.

Note
A sub-array is a contiguous range of elements present in the array beginning at a certain index i and ending at an index $j(j \geq i)$.

Input
The first line of the input contains a single integer n — the size of the array.
The next line in the input contains n integers , the elements of the array.
Output
Print a single integer corresponding to the count of perfect subarrays.

Constraints

- for $subtask1$:
  $1 \leq n \leq 10^3$
  $-10^9 \leq ar[i] \leq 10^9$
- for $subtask2$:
  $1 \leq n \leq 10^6$
  $-10^9 \leq ar[i] \leq 10^9$

Sample Test Case

Input	Output
3 3 2 -1	5

Input	Output
5 4 2 1 -5 3	9