Positive Sum Interval

A positive sum interval (PSI) of an array is a contiguous subsequence of the array such that the sum of all elements in the subsequence is positive. For example, {2, -2, 3, -2} is a PSI of the array with the numbers {1, 2, -2, 3, -2, -2, 3}, but {2, -2, 3, -2, -2} is not.

There are many applications of PSI, but that is not our main focus. In this problem, however, you are interested in counting how many PSIs are ther in a given array of integer.

Input

The first line contains an integer N, the number of elements in the array. The second line contains N integers, separated by a single space. All of the integers are in the interval [-10000, 10000].

Output

Print the number of PSIs in the given array of integers.

Sample Input 1	Sample Output 1
4 1 -2 3 -2	5

Sample Input 2	Sample Output 2
7 1 2 -2 3 -2 -2 3	16

Explanation

For sample input 1, the PSIs are {1}, {1, -2, 3}, {-2, 3}, {3}, {3, -2}.

For sample input 2, you are encouraged to trace them on your own.

Skeleton

You are given the skeleton file `Psi.java`.

Notes

1. You are free to use anything to solve this problem.
2. To pass all test cases on CodeCrunch, your code needs to run in O(N lg N) or faster.
3. If you have coded the O(N lg N) solution but still couldn't pass all test-cases due to WA[1] and not TLE, this cryptic message might help: "If it is not long enough, we should make it long, or else it will go with the flow".

Hints

1. O(N^3) solution: brute-force
2. O(N^2) solution: smart brute-force
3. O(N lg N) solution: do a merge-sort routine with some modifications

[1] See the ICPC problem (sit-in lab 2 – morning session) for the explanation about the terms "WA" and "TLE".