Exquisite Subarrays
Input: *standard input*
Output: *standard output*

Time limit	Memory limit
1 second	128 MB

Statement

You are given an array a of N integers from 0 to 9, $a_1, a_2, \ldots, a_N$. An exquisite subarray is a contiguous sequence of elements in the subarray, such that the sum of the elements is equal to the number of elements in the subarray.

For example, the array $[1, 2, 0]$ has three exquisite subarrays, $[1, 2, 0]$, $[2, 0]$ and $[1]$.

Find the number of exquisite subarrays in a.

Input

The first line of input contains one integer N, the number of elements in a.

The second line contains N space separated integers, the i^{th} of which is the value of a_i.

Output

Print a single integer, the number of exquisite subarrays in a.

As the number of exquisite subarrays can be quite large, it is recommended to use 64 bit integers, that is, long longs in C++

Sample Input 1

```
3
1 2 0
```

Sample Output 1

```
3
```

Sample Input 1

```
5
1 1 0 1 1
```

Sample Output 1

```
6
```

Explanation

For sample input 1, the array $[1, 2, 0]$ has 3 exquisite subarrays, as shown in the statement

For sample input 2, the array $[1, 1, 0, 1, 1]$ has 6 exquisite subarrays: 4 occurrences of $[1]$ and 2 occurrences of $[1, 1]$. Sample input 2 also corresponds to an instance of a test case in subtask 2.

Constraints

- $1 \leq N \leq 100000$
- $0 \leq a_i \leq 9$

Subtasks

- For Subtask 1 (30 marks), $N \leq 1000$.
- For Subtask 2 (30 marks), a_i is either 0 or 1 for all i.
- For Subtask 3 (40 marks), no further constraints apply.