

B. Magical Array

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Valery is very interested in magic. Magic attracts him so much that he sees it everywhere. He explains any strange and weird phenomenon through intervention of supernatural forces. But who would have thought that even in a regular array of numbers Valera manages to see something beautiful and magical.

Valera absolutely accidentally got a piece of ancient parchment on which an array of numbers was written. He immediately thought that the numbers in this array were not random. As a result of extensive research Valera worked out a wonderful property that a magical array should have: an array is defined as magic if its **minimum and maximum coincide**.

He decided to share this outstanding discovery with you, but he asks you for help in return. Despite the tremendous intelligence and wit, Valera counts very badly and so you will have to complete his work. All you have to do is count the number of magical subarrays of the original array of numbers, written on the parchment. Subarray is defined as **non-empty sequence of consecutive elements**.

Input

The first line of the input data contains an integer n ($1 \leq n \leq 10^5$). The second line contains an array of original integers a_1, a_2, \dots, a_n ($-10^9 \leq a_i \leq 10^9$).

Output

Print on the single line the answer to the problem: the amount of subarrays, which are magical.

Please do not use the `%lld` specifier to read or write 64-bit numbers in C++. It is recommended to use `cin`, `cout` streams (you can also use the `%I64d` specifier).

Examples

input
4 2 1 1 4
output
5

input
5 -2 -2 -2 0 1
output
8

Note

Notes to sample tests:

Magical subarrays are shown with pairs of indices $[a;b]$ of the beginning and the end.

In the first sample: $[1;1]$, $[2;2]$, $[3;3]$, $[4;4]$, $[2;3]$.

In the second sample: $[1;1]$, $[2;2]$, $[3;3]$, $[4;4]$, $[5;5]$, $[1;2]$, $[2;3]$, $[1;3]$.