**Odd-Even Subarrays**

You are given an array *A* of *N* positive integer values. A subarray of this array is called Odd-Even subarray if the number of odd integers in this subarray is equal to the number  of even integers in this subarray.

Find the number of Odd-Even subarrays for the given array.

**Input Format**:  
The input consists of two lines.

First line denotes *N* - size of array.  
Second line contains *N* space separated positive integers denoting the elements of array *A*.

**Output Format**:  
Print a single integer, denoting the number of Odd-Even subarrays for the given array.

**Constraints**:

* 1≤N≤2×105
* 1≤A[i]≤109

**SAMPLE INPUT**

4

1 2 1 2

**SAMPLE OUTPUT**

4

**Explanation**

Let A[i..j] denotes the subarray of *A* starting at index *i* and ending at index *j*.

The four subarrays in which number of odd integers are equal to number of even integers are:

A[1..2]=[1,2] contains one odd and one even integer

A[2..3]=[2,1] contains one odd and one even integer

A[3..4]=[1,2] contains one odd and one even integer

A[1..4]=[1,2,1,2] contains two odd and two even integers

#include<bits/stdc++.h>

#define int long long int

using namespace std;

main()

{

int n;

cin>>n;

int a[n],i;

for(i=0;i<n;i++)

cin>>a[i];

for(i=0;i<n;i++)

{

if(a[i]%2)

a[i]=1;

else

a[i]=-1;

}

map<int,int> m;

int curr\_sum=0;

int res=0;

for(i=0;i<n;i++)

{

curr\_sum+=a[i];

if(curr\_sum==0)

res++;

res+=m[curr\_sum];

m[curr\_sum]++;

}

cout<<res;

}