**Similar Chocolates**

Attempted by: **168**

/

Accuracy: **54%**

/

Maximum Score: **20**

/

1 Vote

Tag(s):

Basics of Hash Tables, Data Structures, Easy, Hash Table, Hash Tables,HashMap, Implementation

**PROBLEM**

**EDITORIAL**

**MY SUBMISSIONS**

**ANALYTICS**

There are N chocolates denoted by array A where A[i] is the length of the i-*th*chocolate. Alice can melt each chocolate and then convert it into a chocolate whose length is any divisor of the number A[i]. So, a chocolate of length A[i] can be converted into X different types of chocolate where X is the count of divisors of the number A[i]. So you need to count the total unordered pair of chocolates such that their X value is same.

**Input Format**  
The first line contains an integer N as input denoting the total number of elements in the array A.  
The next line contains N space-separated integers that denote the elements of the array A.  
  
**Output Format**  
In the output, print the total number of ways as mentioned in the statement.  
  
**Constraints**  
1≤N≤105  
1≤A[i]≤106

**SAMPLE INPUT**

3

2 3 4

**SAMPLE OUTPUT**

1

**Explanation**

There is only one possible value i.e. to pick the chocolates 2and 3 as both of them have 2 divisors hence their X value is same.

**Time Limit:**1.0 sec(s) for each input file.

**Memory Limit:**256 MB

**Source Limit:**1024 KB

**Marking Scheme:**Marks are awarded when all the testcases pass.

**Allowed Languages:**Bash, C, C++, C++14, Clojure, C#, D, Erlang, F#, Go, Groovy, Haskell, Java, Java 8, JavaScript(Rhino), JavaScript(Node.js), TypeScript, Julia, Kotlin, Lisp, Lisp (SBCL), Lua, Objective-C, OCaml, Octave, Pascal, Perl, PHP, Python, Python 3, R(RScript), Racket, Ruby, Rust, Scala, Swift, Swift-4.1, Visual Basic

<https://www.hackerearth.com/practice/data-structures/hash-tables/basics-of-hash-tables/practice-problems/algorithm/notebook-pages-dbad75a5/description/>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

class Program

{

static int CantDivisores(int n)

{

int sqr = (int)Math.Sqrt(n);

int cont = 0;

for(int i=1; i<=sqr; i++)

{

if(n%i==0)

{

cont++;

}

}

cont \*= 2;

if (sqr \* sqr == n) cont--;

return cont;

}

static int ObtenerParesIguales(int[] arr)

{

arr = arr.Distinct().ToArray();

Dictionary<int, int> dic =

new Dictionary<int, int>();

for (int i = 0; i < arr.Length; i++)

{

int cant\_div = CantDivisores(arr[i]);

if (dic.ContainsKey(cant\_div)) dic[cant\_div]++;

else dic[cant\_div] = 1;

}

int ans = 0;

foreach (KeyValuePair<int, int> kvp in dic)

{

if (kvp.Value > 1)

{

ans += (kvp.Value \* (kvp.Value - 1)) / 2;

}

}

return ans;

}

static void Main(string[] args)

{

int n = int.Parse(Console.ReadLine());

int[] arr = Array.ConvertAll(Console.ReadLine().Trim().Split(' '), e => int.Parse(e));

Console.WriteLine(ObtenerParesIguales(arr));

Console.ReadLine();

}

}

}