**Gemstones**

https://hr-avatars.s3.amazonaws.com/bbc82ad9-3904-4f41-8ea5-813675b88ef0/150x150.png**by [darkshadows](https://www.hackerrank.com/darkshadows)**

**Problem Statement**

John has discovered various rocks. Each rock is composed of various elements, and each element is represented by a lower-case Latin letter from 'a' to 'z'. An element can be present multiple times in a rock. An element is called a *gem-element* if it occurs at least once in each of the rocks.

Given the list of N rocks with their compositions, display the number of gem-elements that exist in those rocks.

**Input Format**

The first line consists of an integer, N, the number of rocks.   
Each of the next N lines contains a rock's composition. Each composition consists of lower-case letters of English alphabet.

**Constraints**   
1≤N≤100   
Each composition consists of only lower-case Latin letters ('a'-'z').   
1≤ *length of each composition* ≤100

**Output Format**

Print the number of gem-elements that are common in these rocks. If there are none, print 0.

**Sample Input**

3

abcdde

baccd

eeabg

**Sample Output**

2

**Explanation**

Only "a" and "b" are the two kinds of gem-elements, since these are the only characters that occur in every rock's composition.

static void Main(string[] args)

{

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

List<List<char>> lista = new List<List<char>>();

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

{

string roca = Console.ReadLine();

//listaRocas.Add(roca);

List<char> actual = new List<char>();

for (int j = 0; j < roca.Length; j++)

{

if (!actual.Contains(roca[j]))

{

actual.Add(roca[j]);

}

}

lista.Add(actual);

}

int cont = 0;

for (int i = 0; i < lista[0].Count; i++)

{

bool esta = true;

for (int j = 0; j < lista.Count; j++)

{

if (!lista[j].Contains(lista[0][i]))

{

esta = false;

}

}

if (esta)

{

cont++;

}

//Console.WriteLine();

}

Console.WriteLine(cont);

Console.ReadLine();

}