Let's call two integers A and B*friends* if each integer from the array divisors is either a divisor of both A and B or neither A nor B. If two integers are *friends*, they are said to be in the same *clan*. How many clans are the integers from 1 to k, inclusive, broken into?

Example

For divisors = [2, 3] and k = 6, the output should be  
numberOfClans(divisors, k) = 4.

The numbers 1 and 5 are friends and form a *clan*, 2 and 4 are friends and form a *clan*, and 3 and 6 do not have friends and each is a *clan* by itself. So the numbers 1 through 6 are broken into 4 clans.

Input/Output

* **[execution time limit] 3 seconds (cs)**
* **[input] array.integer divisors**

A non-empty array of positive integers.

*Guaranteed constraints:*  
2 ≤ divisors.length < 10,  
1 ≤ divisors[i] ≤ 10.

* **[input] integer k**

A positive integer.

*Guaranteed constraints:*  
5 ≤ k ≤ 10.

* **[output] integer**

**[C#] Syntax Tips**

// Prints help message to the console

// Returns a string

**string** **helloWorld**(**string** name) {

Console.Write("This prints to the console when you Run Tests");

**return** "Hello, " + name;

}

<https://app.codesignal.com/arcade/code-arcade/mirror-lake/BLbGNY3kEcvKjBCFC/solutions>

using System;

using System.Collections.Generic;

using System.Text;

namespace ConsoleApp12

{

class Program

{

static string obtenerDivisores(int[] divisores, int n)

{

string div = "";

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

{

if (n % divisores[i] != 0)

{

div += divisores[i].ToString();

//return false;

}

}

return div;

}

static int numberOfClans(int[] divisors, int k)

{

Dictionary<string, int> clanes = new Dictionary<string, int>();

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

{

string divs = obtenerDivisores(divisors,i);

if (clanes.ContainsKey(divs))

{

clanes[divs]++;

}

else

{

clanes[divs] = 1;

}

}

return clanes.Count;

}

static void Main(string[] args)

{

int[] divisors = {2, 3};

int k = 6;

Console.WriteLine(numberOfClans(divisors, k));

Console.ReadLine();

}

}

}