You are given a rational float number n. Return nas an [irreducible fraction](https://en.wikipedia.org/wiki/Irreducible_fraction).

**Example**

For n = 0.5, the output should be  
ToFraction(n) = [1, 2].

0.5 = 5 / 10 = 1 / 2.

**Input/Output**

* **[time limit] 3000ms (cs)**
* **[input] float n**

*Constraints:*  
0 < n ≤ 1000.

* **[output] array.integer**

Array of two elements representing a fraction, where the first element is a numerator, and the second one is the denominator.

<https://codefights.com/challenge/8WufRbNMfy4Hh7Rbj/main>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication2

{

class Program

{

static int[] ToFraction(double n)

{

string nstring = n.ToString();

int indicePunto = Array.IndexOf(nstring.ToCharArray(),',');

if (indicePunto == -1)

{

return new int[] { (int) n, 1 };

}

string ent = nstring.Substring(0,indicePunto );

string dec = nstring.Substring(indicePunto+1 , nstring.Length - indicePunto -1 );

string num = (ent + dec);

long den = (long)Math.Pow(10, num.Length - indicePunto);

long numerador = long.Parse(num);

long denominador = den;

int[] primos = {2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41,

43, 47, 53, 59, 61, 67, 71, 73, 79, 83,

89, 97, 101, 103, 107, 109, 113, 127,

131, 137, 139, 149, 151, 157, 163, 167,

173, 179, 181, 191, 193, 197, 199, 211,

223, 227, 229, 233, 239, 241, 251, 257,

263, 269, 271, 277, 281, 283, 293, 307,

311, 313, 317, 331, 337, 347, 349, 353,

359, 367, 373, 379, 383, 389, 397, 401,

409, 419, 421, 431, 433, 439, 443, 449,

457, 461, 463, 467, 479, 487, 491, 499,

503, 509, 521, 523, 541, 547, 557, 563,

569, 571, 577, 587, 593, 599, 601, 607,

613, 617, 619, 631, 641, 643, 647, 653,

659, 661, 673, 677, 683, 691, 701, 709,

719, 727, 733, 739, 743, 751, 757, 761,

769, 773, 787, 797, 809, 811, 821, 823,

827, 829, 839, 853, 857, 859, 863, 877,

881, 883, 887, 907, 911, 919, 929, 937,

941, 947, 953, 967, 971, 977, 983, 991,

997,1009 };

int i=0;

while (i < primos.Length)

{

if (numerador % primos[i] == 0 && denominador % primos[i] == 0)

{

numerador /= primos[i];

denominador /= primos[i];

}

else

{

i++;

}

if (i == primos.Length)

{

break;

}

if (primos[i] > numerador || primos[i] > denominador)

{

break;

}

}

return new int[] {(int)numerador, (int)denominador };

}

static void Main(string[] args)

{

// int[] res = ToFraction(999.008);

int[] res = ToFraction(734.2384);

//int[] res = ToFraction(0.6);

Console.WriteLine(res[0] + " " + res[1]);

Console.ReadLine();

}

}

}

----------SOLUCION CON MAXIMO COMUN DIVISOR----------------------------

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

static int GCD(int a, int b)

{

if (a == 0) return b;

return GCD(b % a, a);

}

static int[] ToFraction(double n)

{

string nstring = n.ToString();

int indicePunto = Array.IndexOf(nstring.ToCharArray(), ',');

if (indicePunto == -1)

{

return new int[] { (int)n, 1 };

}

string ent = nstring.Substring(0, indicePunto);

string dec = nstring.Substring(indicePunto + 1, nstring.Length - indicePunto - 1);

string num = (ent + dec);

int den = (int)Math.Pow(10, num.Length - indicePunto);

int numerador = int.Parse(num);

int denominador = den;

int gcd = GCD(numerador, denominador);

numerador /= gcd;

denominador /= gcd;

return new int[] { numerador, denominador };

}

static void Main(string[] args)

{

int[] f = ToFraction(0.5);

Console.WriteLine(f[0] + " " + f[1]);

Console.ReadLine();

}

}

}