**Common Denominators**

4254587% of 616416of 4,155[g964](https://www.codewars.com/users/g964)

C#

* [TRAIN](https://www.codewars.com/kata/common-denominators/train/csharp)
* [NEXT KATA](https://www.codewars.com/trainer/csharp)

Details

[Solutions](https://www.codewars.com/kata/common-denominators/solutions/csharp)

[Forks (9)](https://www.codewars.com/kata/common-denominators/forks/csharp)

[Discourse (239)](https://www.codewars.com/kata/common-denominators/discuss/csharp)

* |
* Share this kata:

Common denominators

You will have a list of rationals in the form

{ {numer\_1, denom\_1} , ... {numer\_n, denom\_n} }

or

[ [numer\_1, denom\_1] , ... [numer\_n, denom\_n] ]

or

[ (numer\_1, denom\_1) , ... (numer\_n, denom\_n) ]

where all numbers are positive ints.

You have to produce a result in the form

(N\_1, D) ... (N\_n, D)

or

[ [N\_1, D] ... [N\_n, D] ]

or

[ (N\_1', D) , ... (N\_n, D) ]

or

{{N\_1, D} ... {N\_n, D}}

depending on the language (See Example tests)

in which D is as small as possible and

N\_1/D == numer\_1/denom\_1 ... N\_n/D == numer\_n,/denom\_n.

Example:

convertFracs [(1, 2), (1, 3), (1, 4)] `shouldBe` [(6, 12), (4, 12), (3, 12)]

**Note for Bash**

input is a string, e.g "2,4,2,6,2,8"

output is then "6 12 4 12 3 12"

<https://www.codewars.com/kata/common-denominators/csharp>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

class Program

{

static long gcd(long a, long b)

{

if (b == 0)

return a;

return gcd(b, a % b);

}

// Returns LCM of array elements

static long findlcm(long[] arr)

{

int n = arr.Length;

// Initialize result

long ans = arr[0];

// ans contains LCM of arr[0], ..arr[i]

// after i'th iteration,

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

ans = (((arr[i] \* ans)) /

(gcd(arr[i], ans)));

return ans;

}

public static string convertFrac(long[,] lst)

{

int n = lst.GetLength(0);

if (n == 0) return "";

long[] denominadores = new long[n];

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

{

denominadores[i] = lst[i, 1];

}

long mcm = findlcm(denominadores);

string res = "";

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

{

//if(denominadores[i] != 0)

res += "(" + ((mcm / denominadores[i]) \* lst[i, 0]) + "," + mcm + ")";

}

return res;

}

static void Main(string[] args)

{

long[,] lista =

new long[,] { { 1, 2 },

{ 1, 3 },

{ 1, 4 } };

Console.WriteLine(convertFrac(lista));

Console.ReadLine();

}

}

}