Given a list of fractions find the index of the largest one (0-based).

**Example**

For numerators = [5, 2, 5] and denominators = [6, 3, 4], the output should be  
maxFraction(numerators, denominators) = 2.

5/4 is the largest fraction, so the answer is 2.

Input/Output

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

Array of integers representing numerators of the fractions.

*Guaranteed constraints:*  
2 ≤ numerators.length ≤ 10,  
1 ≤ numerators[i] ≤ 15.

* **[input] array.integer denominators**

Array of integers of the same length as numerators representing denominators where the ithfraction equals to numerators[i] / denominators[i].

*Guaranteed constraints:*  
denominators.length = numerators.length,  
1 ≤ denominators[i] ≤ 15.

* **[output] integer**
  + Index of the largest fraction assuming that none of the fractions are equal.

**[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/challenge/6hgcY4YgsTW2GugT5>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.IO;

namespace ConsoleApp1

{

class Program

{

static int maxFraction(int[] numerators, int[] denominators)

{

double frac = numerators[0]/denominators[0];

double max\_frac = frac;

int max\_ind = 0;

for(int i =1; i<numerators.Length; i++)

{

frac = (double) numerators[i] / denominators[i];

//max\_frac = Math.Max(max\_frac, frac);

if(frac > max\_frac)

{

max\_frac = frac;

max\_ind = i;

}

}

return max\_ind;

}

static void Main(string[] args)

{

int[] numerators = { 5, 2, 5 };

int[] denominators = { 6, 3, 4 };

Console.WriteLine(maxFraction(numerators, denominators));

Console.ReadLine();

}

}

}