**7 kyu**

**Simple Fun #69: Are Equally Strong?**

0283% of 8017 of219[myjinxin2015](https://www.codewars.com/users/myjinxin2015)

C#

* [TRAIN AGAIN](https://www.codewars.com/kata/simple-fun-number-69-are-equally-strong/train/csharp)
* [NEXT KATA](https://www.codewars.com/trainer/csharp)

Details

[Solutions](https://www.codewars.com/kata/simple-fun-number-69-are-equally-strong/solutions/csharp)

[Discourse (6)](https://www.codewars.com/kata/simple-fun-number-69-are-equally-strong/discuss/csharp)

* Add to Collection
* |
* Share this kata:

**Task**

Call two arms equally strong if the heaviest weights they each are able to lift are equal.

Call two people equally strong if their strongest arms are equally strong (the strongest arm can be both the right and the left), and so are their weakest arms.

Given your and your friend's arms' lifting capabilities find out if you two are equally strong.

**Example**

For yourLeft = 10, yourRight = 15, friendsLeft = 15 and friendsRight = 10, the output should be true;

For yourLeft = 15, yourRight = 10, friendsLeft = 15 and friendsRight = 10, the output should be true;

For yourLeft = 15, yourRight = 10, friendsLeft = 15 and friendsRight = 9, the output should be false.

**Input/Output**

* [input] integer yourLeft

A non-negative integer representing the heaviest weight you can lift with your left arm.

* [input] integer yourRight

A non-negative integer representing the heaviest weight you can lift with your right arm.

* [input] integer friendsLeft

A non-negative integer representing the heaviest weight your friend can lift with his or her left arm.

* [input] integer friendsRight

A non-negative integer representing the heaviest weight your friend can lift with his or her right arm.

* [output] a boolean value

<https://www.codewars.com/kata/simple-fun-number-69-are-equally-strong/csharp>

public static bool AreEquallyStrong(int YourLeft, int YourRight, int FriendsLeft, int FriendsRight)

{

//coding and coding..

int[] friend = { YourLeft, YourRight };

int[] you = { FriendsLeft, FriendsRight };

Array.Sort(friend);

Array.Sort(you);

return friend[0] == you[0] && friend[1] == you[1];

}

**namespace myjinxin**

**{**

**using System;**

**using System.Linq;**

**public class Kata**

**{**

**public bool AreEquallyStrong(int yourLeft, int yourRight, int friendsLeft, int friendsRight)**

**{**

**int[]yourPower = {yourLeft, yourRight};**

**int[]friendsPower = {friendsLeft, friendsRight};**

**return yourPower.Max()==friendsPower.Max() && yourPower.Min()==friendsPower.Min();**

**}**

**}**

**}**

**namespace myjinxin**

**{**

**using System;**

**public class Kata**

**{**

**public bool AreEquallyStrong(int YourLeft, int YourRight, int FriendsLeft, int FriendsRight)**

**{**

**return (YourLeft < YourRight) == (FriendsLeft < FriendsRight)**

**? YourLeft == FriendsLeft && YourRight == FriendsRight**

**: YourLeft == FriendsRight && YourRight == FriendsLeft;**

**}**

**}**

**}**