**6 kyu**

**Triple trouble**

3084385% of 1,1991,200 of8,724[joh\_pot](https://www.codewars.com/users/joh_pot)

C#

* [TRAIN AGAIN](https://www.codewars.com/kata/triple-trouble-1/train/csharp)
* [NEXT KATA](https://www.codewars.com/trainer/csharp)

Details

[Solutions](https://www.codewars.com/kata/triple-trouble-1/solutions/csharp)

[Forks (11)](https://www.codewars.com/kata/triple-trouble-1/forks/csharp)

[Discourse (160)](https://www.codewars.com/kata/triple-trouble-1/discuss/csharp)

* Add to Collection
* |
* Share this kata:

Write a function

TripleDouble(long num1, long num2)

which takes numbers num1 and num2 and returns 1 if there is a straight triple of a number at any place in num1and also a straight double of the **same** number in num2.

If this isn't the case, return 0

Examples

TripleDouble(451999277, 41177722899) == 1 // num1 has straight triple 999s and

// num2 has straight double 99s

TripleDouble(1222345, 12345) == 0 // num1 has straight triple 2s but num2 has only a single 2

TripleDouble(12345, 12345) == 0

TripleDouble(666789, 12345667) == 1

<https://www.codewars.com/kata/triple-trouble-1/csharp>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

class Program

{

public static int TripleDouble(long num1, long num2)

{

string a = num1.ToString();

string b = num2.ToString();

HashSet<string> hash\_a = new HashSet<string>();

for (int i = 0; i + 2 < a.Length; i++)

{

string subs = a.Substring(i, 3);

if (subs.Distinct().ToArray().Length == 1)

{

hash\_a.Add(subs.Substring(1));

}

}

for (int i = 0; i + 1 < b.Length; i++)

{

if (hash\_a.Contains(b.Substring(i, 2)))

{

return 1;

}

}

return 0;

}

static void Main(string[] args)

{

Console.WriteLine(TripleDouble(10560002, 100));

Console.ReadLine();

}

}

}