Numbers x and y (x ≠ y) are called *friendly* if the sum of proper divisors of x is equal to y, and the other way round.

Given two integers x and y, your task is to check whether they are *friendly* or not.

**Example**

For x = 220 and y = 284, the output should be  
friendly\_numbers(x, y) = "Yes".

The proper divisors of 220 are 1, 2, 4, 5, 10, 11, 20, 22, 44, 55 and 110, which add up to 284; and the proper divisors of 284 are 1, 2, 4, 71 and 142, which add up to 220.

**Input/Output**

* **[time limit] 3000ms (cs)**
* **[input] integer x**

*Constraints:*  
1 ≤ x ≤ 105.

* **[input] integer y**

*Constraints:*  
1 ≤ y ≤ 105.

* **[output] string**

"Yes" if x and y are *friendly* and "No" otherwise.

<https://codefights.com/challenge/zeS6of248AhuJB3xM>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication2

{

class Program

{

static string friendly\_numbers(int x, int y)

{

int sum\_proper\_x = 0;

for (int i = 1; i <= x / 2; i++)

{

if (x % i == 0)

{

sum\_proper\_x += i;

}

}

int sum\_proper\_y = 0;

for (int i = 1; i <= y / 2; i++)

{

if (y % i == 0)

{

sum\_proper\_y += i;

}

}

if (x!= y && sum\_proper\_x == y && sum\_proper\_y == x)

{

return "Yes";

}

return "No";

}

static void Main(string[] args)

{

Console.WriteLine(friendly\_numbers(6,6));

Console.ReadLine();

}

}

}