Given integers l and r, find the number of trailing zeros in the decimal representation of l! \* (l + 1)! \* ... \* r! (the exclamation mark means [factorial](keyword://factorial)).

**Example**

For l = 4 and r = 10, the output should be  
factorialsProductTrailingZeros(l, r) = 7.

**Input/Output**

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

*Guaranteed constraints:*  
4 ≤ l ≤ r.

* **[input] integer r**

*Guaranteed constraints:*  
l ≤ r ≤ 35.

* **[output] integer**

**[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://codefights.com/challenge/E5WizNgcx4oPRwYFE/solutions>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

public static int trailingZeroes(int n)

{

int r = 0;

while (n > 0)

{

n /= 5;

r += n;

}

return r;

}

static int factorialsProductTrailingZeros(int l, int r)

{

int sum = 0;

for (int i = l; i <= r; i++)

{

sum += trailingZeroes(i);

}

return sum;

}

static void Main(string[] args)

{

Console.WriteLine(factorialsProductTrailingZeros(4,10));

Console.ReadLine();

}

}

}

static int factorialsProductTrailingZeros(int l, int r)

{

int s = 0;

for (int i = l; i <= r; i++)

{

int c = i;

while (c > 0)

{

c /= 5;

s += c;

}

}

return s;

}