Define *l-r-segment number*as the number formed by concatenating all the digits from l to r in ascending order.

Given l and r (l ≤ r), return the *l-r-segment number*.

Example

For l = 2 and r = 4, the output should be  
lrSegmentNumber(l, r) = 234.

Input/Output

* **[execution time limit] 3 seconds (cs)**
* **[input] integer l**

*Guaranteed constraints:*  
1 ≤ l ≤ r.

* **[input] integer r**

*Guaranteed constraints:*  
l ≤ r ≤ 9.

* **[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://app.codesignal.com/challenge/ptsMnLDpgLRo3dbCB>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

class Program

{

//public static int SubarraySum(int[] nums, int k)

//{

// int[][] sumas = new int[nums.Length][];

// for (int i = 0; i < nums.Length; i++)

// {

// sumas[i] = new int[nums.Length];

// int s = 0;

// for (int j = 0; j < nums.Length; j++)

// {

// s += nums[i];

// sumas[i][j] = s;

// }

// }

// for(int i =0; i<sumas.Length; i++)

// {

// for(int j =0; j<sumas.Length; j++)

// {

// Console.Write( sumas[i][j] + " ");

// }

// Console.WriteLine();

// }

// return 0;

//}

static int lrSegmentNumber(int l, int r)

{

int res = 0;

//for (int i = r; i >= l; i--)

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

{

res \*= 10;

res += i;

}

return res;

}

static void Main(string[] args)

{

//int[] nums = { 1, 1, 1 };

//SubarraySum(nums, 3);

Console.WriteLine(lrSegmentNumber(2,5));

Console.ReadLine();

}

}

}