A little boy is studying arithmetics. He has just learned how to add two integers, written one below another, column by column. But he always forgets about the important part - carrying.

Given two integers, find the result which the little boy will get.

*Note: the boy used [this](https://www.mathsisfun.com/numbers/addition-column.html) site as the source of knowledge, feel free to check it out too if you are not familiar with column addition*.

**Example**

For param1 = 456 and param2 = 1734, the output should be  
additionWithoutCarrying(param1, param2) = 1180.

456

1734

+ \_\_\_\_

1180

The little boy goes from right to left:

* 6 + 4 = 10 but the little boy forgets about1 and just writes down 0 in the last column
* 5 + 3 = 8
* 4 + 7 = 11 but the little boy forgets about the leading 1 and just writes down 1 under4 and 7.
* There is no digit in the first number corresponding to the leading digit of the second one, so the little boy imagines that0 is written before 456. Thus, he gets 0 + 1 = 1.

**Input/Output**

* **[time limit] 3000ms (cs)**
* **[input] integer param1**

A non-negative integer.

*Constraints:*  
0 ≤ param1 ≤ 99999.

* **[input] integer param2**

A non-negative integer.

*Constraints:*  
0 ≤ param2 ≤ 59999.

* **[output] integer**

The result that the little boy will get.

<https://codefights.com/arcade/code-arcade/loop-tunnel/xzeZqCQjpfDJuN72S>

static int additionWithoutCarrying(int param1, int param2)

{

string p1 = param1.ToString();

string p2 = param2.ToString();

if (p1.Length > p2.Length)

{

string ceros = new string('0', p1.Length - p2.Length);

p2 = ceros + p2;

}

else if (p1.Length < p2.Length)

{

string ceros = new string('0', p2.Length - p1.Length);

p1 = ceros + p1;

}

string concat = "";

for (int i = 0; i < p1.Length; i++)

{

concat += ((int.Parse(p1[i].ToString()) + int.Parse(p2[i].ToString()))%10);

}

return int.Parse(concat);

}