You work in a company that prints and publishes books. You are responsible for designing the page numbering mechanism in the printer. You know how many digits a printer can print with the leftover ink. Now you want to write a function to determine what the last page of the book is that you can number given the current page and numberOfDigits left. A page is considered numbered if it has the full number printed on it (e.g. if we are working with page 102 but have ink only for two digits then this page will not be considered numbered).

It's guaranteed that you can number the currentpage, and that you can't number the last one in the book.

**Example**

* For current = 1 and numberOfDigits = 5, the output should be  
  pagesNumberingWithInk(current, numberOfDigits) = 5.

The following numbers will be printed: 1, 2, 3, 4, 5.

* For current = 21 and numberOfDigits = 5, the output should be  
  pagesNumberingWithInk(current, numberOfDigits) = 22.

The following numbers will be printed: 21, 22.

* For current = 8 and numberOfDigits = 4, the output should be  
  pagesNumberingWithInk(current, numberOfDigits) = 10.

The following numbers will be printed: 8, 9, 10.

**Input/Output**

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

A positive integer, the number on the current page which is not yet printed.

*Constraints:*  
1 ≤ current ≤ 1000.

* **[input] integer numberOfDigits**

A positive integer, the number of digits which your printer can print.

*Constraints:*  
1 ≤ numberOfDigits ≤ 1000.

* **[output] integer**

The last printed page number.

<https://codefights.com/arcade/code-arcade/labyrinth-of-nested-loops/pdw3izd7SpMTBJqSy>

static int pagesNumberingWithInk(int current, int numberOfDigits)

{

int contDig = 0;

int i = current;

for ( ; ; i++)

{

contDig += i.ToString().Length;

if (contDig + (i + 1).ToString().Length > numberOfDigits)

{

break;

}

}

return i;

}