Define an integer's *roundness* as the number of trailing zeroes in it.

Given an integer n, check if it's possible to increase n's roundness by swapping some pair of its digits.

**Example**

* For n = 902200100, the output should be  
  increaseNumberRoundness(n) = true.

For instance, one may swap the leftmost 0with 1.

* For n = 11000, the output should be  
  increaseNumberRoundness(n) = false.

**Input/Output**

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

A positive integer.

*Constraints:*  
100 ≤ n ≤ 109.

* **[output] boolean**

true if it's possible to increase n's roundness, false otherwise.

<https://codefights.com/arcade/code-arcade/loop-tunnel/KLbRMcWhaZi3dvYH5>

static bool increaseNumberRoundness(int n)

{

string ns = n.ToString();

for (int i = 0; i + 1 < ns.Length; i++)

{

if (ns[i] == '0' && ns[i+1] != '0')

{

return true;

}

}

return false;

}