We want to turn the given integer into a number that has only one non-zero digit using a tail rounding approach. This means that at each step we take the last non 0 digit of the number and round it to 0 or to 10. If it's less than 5 we round it to 0 if it's larger than or equal to 5 we round it to 10 (rounding to 10 means increasing the next significant digit by 1).

**Example**

* For value = 15, the output should be  
  rounders(value) = 20;
* For value = 1234, the output should be  
  rounders(value) = 1000.

1234 -> 1230 -> 1200 -> 1000.

* For value = 1445, the output should be  
  rounders(value) = 2000.

1445 -> 1450 -> 1500 -> 2000.

**Input/Output**

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

A positive integer.

*Constraints:*  
1 ≤ value ≤ 108.

* **[output] integer**

The rounded number.

<https://codefights.com/arcade/code-arcade/loop-tunnel/H5PP5MXvYvWXxTytH>

static int rounders(int value)

{

string vs = value.ToString();

string res = "";

int acarreo = 0;

for (int i = vs.Length - 1; i >= 0; i--)

{

if (i == 0)

{

res = (int.Parse(vs[i].ToString()) + acarreo).ToString() + res;

}

else

{

if (int.Parse(vs[i].ToString()) + acarreo >= 5)

{

acarreo = 1;

int dig = int.Parse(vs[i].ToString()) + acarreo;

res += "0";

}

else

{

acarreo = 0;

res += "0";

}

}

}

return int.Parse( res );

}