A Rule of Divisibility by 7

A number m of the form 10x + y is divisible by 7 if and only if x − 2y is divisible by 7. In other words, subtract twice the last digit from the number formed by the remaining digits. Continue to do this until a number known to be divisible or not by 7 is obtained; you can stop when this number has *at most* 2 digits because you are supposed to know if a number of at most 2 digits is divisible by 7 or not.

The original number is divisible by 7 if and only if the last number obtained using this procedure is divisible by 7.

Examples:

1 - m = 371 -> 37 − (2×1) -> 37 − 2 = 35 ; thus, since 35 is divisible by 7, 371 is divisible by 7.

The number of steps to get the result is 1.

2 - m = 1603 -> 160 - (2 x 3) -> 154 -> 15 - 8 = 7 and 7 is divisible by 7.

3 - m = 372 -> 37 − (2×2) -> 37 − 4 = 33 ; thus, since 33 is not divisible by 7, 372 is not divisible by 7.

The number of steps to get the result is 1.

4 - m = 477557101->47755708->4775554->477547->47740->4774->469->28 and 28 is divisible by 7, so is 477557101.

The number of steps is 7.

#Task: Your task is to return to the function seven(m) (m integer >= 0) an array (or a pair, depending on the language) of numbers, the first being the *last*number m with at most 2 digits obtained by your function (this last m will be divisible or not by 7), the second one being the number of steps to get the result.

seven(371) should return [35, 1] seven(1603) should return [7, 2] seven(477557101) should return [28, 7]

using System;

using System.Linq;

public class DivSeven

{

public static long count = 0;

public static long[] Seven(long m)

{

// your code

long[] numbersDivisibleBySeven = {7,14,21,28,35,42,49,56,63,70,77,84,91,98};

if(m <= 98)

{

if(numbersDivisibleBySeven.Contains(m))

{

return new long[] {m, count};

}

else

{

return new long[] {0, 0};

}

}

else

{

long lastDigit = m % 10;

long tenX = m - lastDigit;

long x = tenX / 10;

long intermediateResult = x - 2 \* lastDigit;

count ++;

return Seven(intermediateResult);

}

}

}