Let's call an integer *unusual* if the sum of its digits is larger than the product of its digits. For example, the numbers 21 and 990 are *unusual*, while the numbers 22 and 991 aren't.

Given an integer a (represented as a string), find the smallest *unusual* integer x such that x ≥ a. Since both x and a can be very large, return the value of x - a.

**Example**

For a = "42", the output should be  
smallestUnusualNumber(a) = 8.

The smallest *unusual* number that is greater than or equal to 42 is 50, and 50 - 42 = 8.

**Input/Output**

* **[time limit] 3000ms (cs)**
* **[input] string a**

A string representing a positive integer without leading zeros.

*Constraints:*  
1 ≤ a.length ≤ 105.

* **[output] integer**

The difference between the smallest *unusual* number that is larger than or equal to a, and a.

<https://codefights.com/tournaments/RrqS57vhTAkt9WjYq/E/solutions>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication3

{

class Program

{

static bool SumaEsMayor(string a)

{

if (a.Contains('0'))

{

return true;

}

int sum = a.Sum(e => e - '0');

//Console.WriteLine(sum);

int prod = 1;

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

{

prod \*= int.Parse(a[i].ToString());

if (prod > sum)

{

return false;

}

}

if (sum > prod)

{

return true;

}

return false;

}

static int smallestUnusualNumber(string a)

{

if (a.Contains('0'))

{

return 0;

}

if (a == "1")

{

return 9;

}

if (SumaEsMayor(a))

{

return 0;

}

return 10 - int.Parse(a[a.Length - 1].ToString());

}

static void Main(string[] args)

{

//string a = "8888888888888888888888888888888";

string a = "22";

Console.WriteLine(smallestUnusualNumber(a));

// Console.WriteLine(SumaEsMayor("123"));

Console.ReadLine();

}

}

}