A step(x) operation works like this: it changes a number x into x - s(x), where s(x) is the sum of x's digits. You like applying functions to numbers, so given the number n, you decide to build a decreasing sequence of numbers: n, step(n), step(step(n)), etc., with 0 as the last element.

Building a single sequence isn't enough for you, so you replace all elements of the sequence with the sums of their digits (s(x)). Now you're curious as to which number appears in the new sequence most often. If there are several answers, return the maximal one.

**Example**

* For n = 88, the output should be  
  mostFrequentDigitSum(n) = 9.
  + Here is the first sequence you built: 88, 72, 63, 54, 45, 36, 27, 18, 9, 0;
  + And here is s(x) for each of its elements: 16, 9, 9, 9, 9, 9, 9, 9, 9, 0.

As you can see, the most frequent number in the second sequence is 9.

* For n = 8, the output should be   
  mostFrequentDigitSum(n) = 8.
  + At first you built the following sequence: 8, 0
  + s(x) for each of its elements is: 8, 0

As you can see, the answer is 8 (it appears as often as 0, but is greater than it).

**Input/Output**

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

*Constraints:*  
1 ≤ n ≤ 105,

* **[output] integer**

The most frequent number in the sequence s(n), s(step(n)), s(step(step(n))), etc.

<https://codefights.com/arcade/code-arcade/mirror-lake/RpoP4Aqa5mbmC8koC>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

static int s(int n)

{

int sum = 0;

while (n > 0)

{

sum += n % 10;

n /= 10;

}

return sum;

}

static int mostFrequentDigitSum(int n)

{

List<int> lista = new List<int>();

lista.Add(n);

do

{

n = n - s(n);

lista.Add(n);

} while (n != 0);

Dictionary<int, int> frec = new Dictionary<int, int>();

for (int i = 1; i < lista.Count; i++)

{

int key = lista[i - 1] - lista[i];

if (frec.ContainsKey(key))

{

frec[key]++;

}

else

{

frec[key] = 1;

}

}

frec[0] = 1;

int maxKey = 0, maxValue =0;

foreach (KeyValuePair<int, int> kvp in frec)

{

// Console.WriteLine(kvp.Key + " " + kvp.Value);

if (kvp.Value > maxValue)

{

maxValue = kvp.Value;

maxKey = kvp.Key;

}

else if (kvp.Value == maxValue)

{

if (kvp.Key > maxKey)

{

maxKey = kvp.Key;

}

}

}

return maxKey;

}

static void Main(string[] args)

{

Console.WriteLine( mostFrequentDigitSum(88) );

Console.ReadLine();

}

}

}