Given a multiple-digit number num, determine its *true value* by performing the following operations:

1. The digit at the ones' place should be multiplied by1. The digit at the tens' place should be multiplied by 2. The digit at the hundreds' place should be multiplied by 3, and so on.
2. Repeat step 1 until the result becomes a one-digit number.

**Example**

For num = 234, the output should be  
TrueValue(num) = 8.

The first iteration produces 4 \* 1 +3 \* 2 + 2 \* 3 = 4 + 6 + 6 = 16. After the second iteration, the number becomes 6 \* 1 + 1 \* 2 = 6 + 2 = 8. Thus, its *true value* is 8.

**Input/Output**

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

*Constraints:*  
10 ≤ num ≤ 2 · 109.

* **[output] integer**

The *true value* of num.

<https://codefights.com/challenge/qqWWuFHbipD4B83Tv/main>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

static int TrueValue(int num)

{

string ns = num.ToString();

while (ns.Length > 1)

{

int sum = 0;

int prod = 1;

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

{

sum += int.Parse(ns[i].ToString()) \* prod++;

}

ns = sum.ToString();

}

return int.Parse(ns);

}

static void Main(string[] args)

{

Console.WriteLine( TrueValue(234));

Console.ReadLine();

}

}

}