Author

[sudo\_su](https://codefights.com/profile/sudo_su)

https://codefights.com/img/coins_new.png2000

You are given a positive integer n either as a number, or in the format "a^b", where a and b are positive integers (possibly with leading zeros).

Your task is to sum all digits in n. Repeat the process until you get a single digit and return it.

**Example**

* tillsingle("4444") = 7  
  4444 => 4 + 4 + 4 + 4 = 16  
  16 => 1 + 6 = 7
* tillsingle("2^9") = 8  
  29 = 512  
  512 => 5 + 1 + 2 = 8
* **[input] string n**

0 ≤ n ≤ 251

* **[output] integer**

The resulting digit.

<https://codefights.com/challenge/etA6tQkKhoNGrMb4r>

static int tillsingle(string n)

{

string[] split = n.Split('^');

long num = 0;

if (split.Length > 1)

{

num = (long)Math.Pow(long.Parse(split[0]), long.Parse(split[1]));

}

else

{

num = long.Parse( n );

}

while (num.ToString().Length > 1)

{

char[] digitos = num.ToString().ToCharArray();

long actual = 0;

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

{

//Console.WriteLine(digitos[i]);

actual += int.Parse(digitos[i].ToString());

}

//Console.WriteLine(actual);

num = actual;

}

// Console.WriteLine(num);

return (int) num;

}