**7 kyu**

**Numbers with this digit inside**

751686% of 504135of 2,512[SteffenVogel\_79](https://www.codewars.com/users/SteffenVogel_79)

C#

* [TRAIN AGAIN](https://www.codewars.com/kata/numbers-with-this-digit-inside/train/csharp)
* [NEXT KATA](https://www.codewars.com/trainer/csharp)

Details

[Solutions](https://www.codewars.com/kata/numbers-with-this-digit-inside/solutions/csharp)

[Forks (3)](https://www.codewars.com/kata/numbers-with-this-digit-inside/forks/csharp)

[Discourse (95)](https://www.codewars.com/kata/numbers-with-this-digit-inside/discuss/csharp)

* Add to Collection
* |
* Share this kata:

You have to search all numbers from inclusive 1 to inclusive a given number x, that have the given digit d in it.  
The value of d will always be 0 - 9.  
The value of x will always be greater than 0.

You have to return as an array

**the count of these numbers,**

**their sum**

**and their product.**

For example:

x = 11

d = 1

->

Numbers: 1, 10, 11

Return: [3, 22, 110]

If there are no numbers, which include the digit, return [0,0,0].   
  
Have fun coding it and please don't forget to vote and rank this kata! :-)   
  
I have created other katas. Have a look if you like coding and challenges.

<https://www.codewars.com/kata/numbers-with-this-digit-inside/csharp>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

class Program

{

public static long[] NumbersWithDigitInside(long x, long d)

{

var intArray = new long[3] { 0, 0, 1 };

var digit = d.ToString();

for (long i = 1; i <= x; i++)

{

string temp = i.ToString();

if (!temp.Contains(digit)) continue;

intArray[0]++;

intArray[1] += i;

intArray[2] \*= i;

}

if (intArray[0] == 0) intArray[2] = 0;

return intArray;

}

static bool Contine(long n, long d)

{

while (n > 0)

{

int dig = (int)n % 10;

if (dig == d) return true;

n /= 10;

}

return false;

}

public static long[] NumbersWithDigitInside(long x, long d)

{

//return new long[0];

long cont = 0;

long sum = 0, prod = 1;

bool flag = false;

for (int i = 1; i <= x; i++)

{

if (Contine(i, d))

{

cont++;

sum += i;

prod \*= i;

flag = true;

}

}

if(flag) return new long[] { cont, sum, prod };

return new long[] { 0, 0, 0 };

}

static void Main(string[] args)

{

long[] l = new long[] { 2, 30, 200 };

long[] res = NumbersWithDigitInside(20, 0);

Console.WriteLine(res[0] + " " + res[1] + " " + res[2]);

Console.ReadLine();

}

}

}