**7 kyu**

**Numbers Which Sum of Powers of Its Digits Is The Same Number**

23593% of 60120 of842[raulbc777](http://www.codewars.com/users/raulbc777)

* C#
* 7.3

Instructions

Output

* Not considering number 1, the integer 153 is the first integer having this property: the sum of the third-power of each of its digits is equal to 153. Take a look: 153 = 1^3 + 5^3 + 3^3 = 1 + 125 + 27 = 153

The next number that experiments this particular behaviour is 370 with the same power.

Write the function eq\_sum\_powdig(), that finds the numbers below a given upper limit hMax that fulfill this property but with different exponents as a power for the digits.

eq\_sum\_powdig(hMax, exp): ----> sequence of numbers (sorted list) (Number one should not be considered).

Let's see some cases:

eq\_sum\_powdig(100, 2) ----> []

eq\_sum\_powdig(1000, 2) ----> []

eq\_sum\_powdig(200, 3) ----> [153]

eq\_sum\_powdig(370, 3) ----> [153, 370]

Enjoy it !!

<http://www.codewars.com/kata/numbers-which-sum-of-powers-of-its-digits-is-the-same-number/train/csharp>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

class Program

{

public static long[] EqSumPowDig(long hmax, int exp)

{

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

// your code

for(long i = 2; i<=hmax; i++)

{

long n = i;

long sum = 0;

while (n>0)

{

sum += (long) Math.Pow (n % 10, exp);

n /= 10;

}

if (sum == i)

{

lista.Add(sum);

}

}

return lista.ToArray();

}

static void Main(string[] args)

{

foreach (int elem in EqSumPowDig(400, 3))

{

Console.Write(elem + " ");

}

Console.ReadLine();

}

}

}