*Emirp* is prime spelled backwards.

A number is *emirp* if both the number itself and the same number written backwards are prime. For instance, 13 and 31 are *emirps*, but 23 is not, since 32 is a composite number.

*Emirps* are hiding in a large integer num, waiting for the next skilled programmer to find them! They are clustered together, so an *emirp* can be formed from a set of consecutive digits of num.

Find the sum of all *emirps* in num. If none can be found, return -1 instead.

**Example**

For num = 13211, the output should be  
emirpMinion(num) = 1350.

In this example, 13, 3, 2, 1321, and 11 are the only *emirps*, and their sum is 13 + 3 + 2 + 11 + 1321 = 1350.

**Input/Output**

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

Large integer where *emirps* are hiding.

*Constraints:*  
10 ≤ num ≤ 2 · 109.

* **[output] integer**

Sum of all *emirps* found in the large integernum.

<https://codefights.com/challenge/52cFreA4YEu3vmLu5/main>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

static bool esPrimo(long n)

{

if (n < 2) return false;

if (n == 2) return true;

if (n % 2 == 0) return false;

long sqr = (long)Math.Sqrt(n);

for (long i = 3; i <= sqr; i += 2)

{

if (n % i == 0)

return false;

}

return true;

}

static bool esEmirp(long n)

{

char[] rev = n.ToString().ToCharArray();

Array.Reverse(rev);

if (esPrimo(long.Parse(new string(rev))) && esPrimo(n))

{

return true;

}

return false;

}

static long emirpMinion(long num)

{

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

bool hayEmirp = false;

long sum = 0;

string ns = num.ToString();

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

{

for (int j = i; j < ns.Length; j++)

{

string subs = ns.Substring(i, j - i + 1);

// Console.WriteLine(subs);

if (esEmirp(long.Parse(subs)))

{

if (subs[0] != '0')

{

sum += long.Parse(subs);

}

hayEmirp = true;

}

}

}

if (hayEmirp)

{

return sum;

}

return -1;

}

static void Main(string[] args)

{

//emirpMinion(70707);

Console.WriteLine(emirpMinion(70707));

Console.ReadLine();

}

}

}