**Closest Gift**

Attempted by: **2108**

/

Accuracy: **89%**

/

Maximum Score: **10**

/

36 Votes

Tag(s):

Very-Easy

**PROBLEM**

**EDITORIAL**

**MY SUBMISSIONS**

**ANALYTICS**

You are on your way to find the gifts. All the gifts lie in your path in a straight line at prime numbers and your house is at 0.Given your current position find the closest gift to your position, and calculate the distance between your current position and gift and tell the distance.

**SAMPLE INPUT**

0

**SAMPLE OUTPUT**

2

**Explanation**

For the no.= 0, the output is 2.

The closest prime number to 0 is 2, so the answer is 2 - 0 = 2.

For number = 11, the output should be 0.

11 is a prime number, so the answer is 11 - 11 = 0.

**Input/Output**

[time limit] 3000ms [input] string number

**Constraints:** 0 ≤ int(number) ≤ 9 · 1014.

[output] integer

**Time Limit:**3,0 sec(s) for each input file.

**Memory Limit:**256 MB

**Source Limit:**1024 KB

**Marking Scheme:**Marks are awarded when all the testcases pass.

**Allowed Languages:**C, C++, C++14, Clojure, C#, D, Erlang, F#, Go, Groovy, Haskell, Java, Java 8, JavaScript(Rhino), JavaScript(Node.js), Julia, Kotlin, Lisp, Lisp (SBCL), Lua, Objective-C, OCaml, Octave, Pascal, Perl, PHP, Python, Python 3, R(RScript), Racket, Ruby, Rust, Scala, Swift, Visual Basic

<https://www.hackerearth.com/fr/practice/math/number-theory/basic-number-theory-1/practice-problems/algorithm/closest-gift/description/>

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 (int i = 3; i <= sqr; i += 2)

{

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

}

return true;

}

static void Main(string[] args)

{

long n = long.Parse(Console.ReadLine());

if (esPrimo(n))

{

Console.WriteLine(0);

}

else

{

long der;

long izq = der = n;

while (izq >= 2 && !esPrimo(izq))

{

izq--;

}

while (!esPrimo(der))

{

der++;

}

if (izq >= 2)

{

Console.WriteLine(Math.Min(n - izq, der - n));

}

else

{

Console.WriteLine(der - n);

}

}

Console.ReadLine();

}

}

}