**6 kyu**

**Steps in Primes**

1532489% of 359191of 1,861[g964](https://www.codewars.com/users/g964)

C++

* [TRAIN AGAIN](https://www.codewars.com/kata/steps-in-primes/train/cpp)
* [NEXT KATA](https://www.codewars.com/trainer/cpp)

Details

[Solutions](https://www.codewars.com/kata/steps-in-primes/solutions/cpp)

[Forks (1)](https://www.codewars.com/kata/steps-in-primes/forks/cpp)

[Discourse (104)](https://www.codewars.com/kata/steps-in-primes/discuss/cpp)

* Add to Collection
* |
* Share this kata:

The prime numbers are not regularly spaced. For example from 2 to 3 the step is 1. From 3 to 5the step is 2. From 7 to 11 it is 4. Between 2 and 50 we have the following pairs of 2-steps primes:

3, 5 - 5, 7, - 11, 13, - 17, 19, - 29, 31, - 41, 43

We will write a function step with parameters:

* g (integer >= 2) which indicates the step we are looking for,
* m (integer >= 2) which gives the start of the search (m inclusive),
* n (integer >= m) which gives the end of the search (n inclusive)

In the example above step(2, 2, 50) will return [3, 5] which is the first pair between 2 and 50 with a 2-steps.

So this function should return the **first** pair of the two prime numbers spaced with a step of g between the limits m, n if these g-steps prime numbers exist otherwise nil or null or None or Nothing or [] or "0, 0" or {0, 0} (depending on the language).

#Examples:

step(2, 5, 7) --> [5, 7] or (5, 7) or {5, 7} or "5 7"

step(2, 5, 5) --> nil or ... or [] in Ocaml or {0, 0} in C++

step(4, 130, 200) --> [163, 167] or (163, 167) or {163, 167}

* **See more examples for your language in "RUN"**
* Remarks:

([193, 197] is also such a 2-steps primes between 130 and 200 but it's not the first pair).

step(6, 100, 110) --> [101, 107] though there is a prime between 101 and 107 which is 103; the pair 101-103 is a 2-step.

#Notes: The idea of "step" is close to that of "gap" but it is not exactly the same. For those interested they can have a look at <http://mathworld.wolfram.com/PrimeGaps.html>.

A "gap" is more restrictive: there must be no primes in between (101-107 is a "step" but not a "gap". Next kata will be about "gaps":-).

For Go: nil slice is expected when there are no step between m and n. Example: step(2,4900,4919) --> nil

<https://www.codewars.com/kata/steps-in-primes/cpp>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

class Program

{

static bool EsPrimo(long n)

{

if (n < 2) return false;

if (n == 2) return true;

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

for (long i = 3; i \* i <= n; i += 2) if (n % i == 0) return false;

return true;

}

public static long[] Step(int g, long m, long n)

{

// your code

long[] res = new long[2];

for (long i = m; i +g<= n; i++)

{

if (EsPrimo(i) && EsPrimo(i + g))

{

res[0] = i;

res[1] = i + g;

return res;

}

}

return null;

}

static void Main(string[] args)

{

// Assert.AreEqual(new int[] { 307, 317 }

foreach (long elem in Step(10, 1300, 1400))

{

Console.Write(elem + " ");

}

Console.ReadLine();

}

}

}

#include <iostream>

#include <stdio.h>

#include <iostream>

#include <utility>

using namespace std;

bool EsPrimo(long n)

{

if (n < 2) return false;

if (n == 2) return true;

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

for (long i = 3; i \* i <= n; i += 2) if (n % i == 0) return false;

return true;

}

std::pair<long long, long long> step(int g, long long m, long long n) {

//long[] res = new long[2];

//pair<long long, long long> p;

for (long i = m; i + g <= n; i++)

{

if (EsPrimo(i) && EsPrimo(i + g)) return std::make\_pair(i, i+g);

}

return std::make\_pair(0,0);

}

int main() {

return 0;

}