**5 kyu**

**Primes in the Last Digits of Huge Numbers**

41100% of 713 of17[raulbc777](https://www.codewars.com/users/raulbc777)

Python

* [TRAIN AGAIN](https://www.codewars.com/kata/primes-in-the-last-digits-of-huge-numbers/train/python)
* [NEXT KATA](https://www.codewars.com/trainer/python)

Details

[Solutions](https://www.codewars.com/kata/primes-in-the-last-digits-of-huge-numbers/solutions/python)

[Discourse (4)](https://www.codewars.com/kata/primes-in-the-last-digits-of-huge-numbers/discuss/python)

* Add to Collection
* |
* Share this kata:

We have the following sequence:

f(0) = 0

f(1) = 1

f(2) = 1

f(3) = 2

f(4) = 4;

f(n) = f(n-1) + f(n-2) - f(n-3) + f(n-4) - f(n-5);

The first term of the sequence that has its last nine digits forms a prime number is the value, 8480150779 (total of 10 digits), and corresponds to the 92-th term, because 480150779 is prime. We can follow this observation in the next terms of the sequence and see the behaviour.

n-th term k-th lastDig prime term value total digits last9Digit isPrime(last9Digit)

92 1 8480150779 10 480150779 True

98 2 35922495169 11 922495169 True

110 3 644603021049 12 603021049 True

122 4 11566931883761 14 931883761 True

134 5 207560170886697 15 170886697 True

Create a function kth\_lastDigPrime(), that receives the value of k as an argument and outputs, the ordinal number that corresponds to the term value and the number formed by the last nine digits. Let's see some cases:

k\_thlastDigPrime(1)--------> [92, 480150779]

k\_thlastDigPrime(2)--------> [98, 922495169]

k\_thlastDigPrime(5)--------> [134, 170886697]

(Advise: Use a fast primality test, Miller Rabin test or similar)

Happy coding!

<https://www.codewars.com/kata/55e61967663140aafb0001d6/solutions/python>

def \_try\_composite(a, d, n, s):

if pow(a, d, n) == 1:

return False

for i in range(s):

if pow(a, 2\*\*i \* d, n) == n-1:

return False

return True # n is definitely composite

def is\_prime(n, \_precision\_for\_huge\_n=16):

if n in \_known\_primes or n in (0, 1):

return True

if any((n % p) == 0 for p in \_known\_primes):

return False

d, s = n - 1, 0

while not d % 2:

d, s = d >> 1, s + 1

# Returns exact according to http://primes.utm.edu/prove/prove2\_3.html

if n < 1373653:

return not any(\_try\_composite(a, d, n, s) for a in (2, 3))

if n < 25326001:

return not any(\_try\_composite(a, d, n, s) for a in (2, 3, 5))

if n < 118670087467:

if n == 3215031751:

return False

return not any(\_try\_composite(a, d, n, s) for a in (2, 3, 5, 7))

if n < 2152302898747:

return not any(\_try\_composite(a, d, n, s) for a in (2, 3, 5, 7, 11))

if n < 3474749660383:

return not any(\_try\_composite(a, d, n, s) for a in (2, 3, 5, 7, 11, 13))

if n < 341550071728321:

return not any(\_try\_composite(a, d, n, s) for a in (2, 3, 5, 7, 11, 13, 17))

# otherwise

return not any(\_try\_composite(a, d, n, s)

for a in \_known\_primes[:\_precision\_for\_huge\_n])

\_known\_primes = [2, 3]

\_known\_primes += [x for x in range(5, 1000, 2) if is\_prime(x)]

def k\_thlastDigPrime(k):

#your code here

lista = []

lista.append(0);

lista.append(1);

lista.append(1);

lista.append(2);

lista.append(4);

t = 0

primo = 0

i=5

#for i in range(5, 500):

while True:

lista.append(lista[i-1]+lista[i-2] - lista[i-3] + lista[i-4] - lista[i-5])

if len( str(lista[i])) >= 9:

#is\_prime(lista[i]):

s = str(lista[i])

l = len(s)

subs = s[l-9:l]

e = int(subs)

#print (lista[i])

if is\_prime(e):

#if e == 480150779:

primo += 1

if primo == k:

return [i+1, e]

#print((i+1) , " => " , e)

i+=1

#print(e)

#print("fin")

#print (lista)

return []

#k\_thlastDigPrime(1)

print (k\_thlastDigPrime(1))

'''

s = "12345678901234"

l = len(s)

subs = s[l-9:l]

print(subs)

'''

'''

for i in range(2,100):

if is\_prime(i):

print(i)

'''