**6 kyu**

**Total Primes**

10095% of2141[Mcoury](https://www.codewars.com/users/Mcoury)

Python

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

Details

[Solutions](https://www.codewars.com/kata/total-primes/solutions/python)

[Forks (1)](https://www.codewars.com/kata/total-primes/forks/python)

[Discourse (21)](https://www.codewars.com/kata/total-primes/discuss/python)

* Add to Collection
* |
* Share this kata:

The number 23 is special in the sense that all of its digits are prime numbers. Furthermore, it's **a prime itself**. There are 4 such numbers between 10 and 100: 23, 37, 53, 73. Let's call these numbers "total primes".

Complete the function that takes a range (a, b) and returns the number of total primes within that range (a <= primes < b). The test ranges go up to 107.

Examples

(10, 100) ==> 4 # 23, 37, 53, 73

(500, 600) ==> 3 # 523, 557, 577

<https://www.codewars.com/kata/total-primes/python>

**from itertools import product**

**def isPrime(n):**

**return n==2 or n%2 and all(n%p for p in range(3,int(n\*\*.5)+1,2))**

**def get\_total\_primes(a, b):**

**low, high = map(len, map(str, (a,b)))**

**return sum( a <= n < b and isPrime(n) for d in range(low,high+1) for n in map(int, map(''.join, product("2357",repeat=d))))**