**6 kyu**

**The Sum Of The Prime Factors Of a Number... What For?**

7295% of 3344 of75[raulbc777](https://www.codewars.com/users/raulbc777)

Python

* [TRAIN AGAIN](https://www.codewars.com/kata/the-sum-of-the-prime-factors-of-a-number-dot-dot-dot-what-for/train/python)
* [NEXT KATA](https://www.codewars.com/trainer/python)

Details

[Solutions](https://www.codewars.com/kata/the-sum-of-the-prime-factors-of-a-number-dot-dot-dot-what-for/solutions/python)

[Forks (1)](https://www.codewars.com/kata/the-sum-of-the-prime-factors-of-a-number-dot-dot-dot-what-for/forks/python)

[Discourse (13)](https://www.codewars.com/kata/the-sum-of-the-prime-factors-of-a-number-dot-dot-dot-what-for/discuss/python)

* Add to Collection
* |
* Share this kata:

Every number may be factored in prime factors.

For example, the number 18 may be factored by its prime factors 2 and 3

18 = 2 . 3 . 3 = 2 . 3²

The sum of the prime factors of 18 is 2 + 3 + 3 = 8

But some numbers like 70 are divisible by the sum of its prime factors:

70 = 2 . 5 . 7 # sum of prime factors = 2 + 5 + 7 = 14

and 70 is a multiple of 14

Of course that primes would fulfill this property, but is obvious, because the prime decomposition of a number, is the number itself and every number is divisible by iself. That is why we will discard every prime number in the results

We are interested in collect the integer positive numbers (non primes) that have this property in a certain range [a, b] (inclusive).

Make the function mult\_primefactor\_sum(), that receives the values a, b as limits of the range [a, b] and a < b and outputs the sorted list of these numbers.

Let's see some cases:

mult\_primefactor\_sum(10, 100) == [16, 27, 30, 60, 70, 72, 84]

mult\_primefactor\_sum(1, 60) == [1, 4, 16, 27, 30, 60]

<https://www.codewars.com/kata/the-sum-of-the-prime-factors-of-a-number-dot-dot-dot-what-for/python>

**import** math

**def** primeFactors(n):

*# Print the number of two's that divide n*

    suma = 0

**while** n % 2 == 0:

*#print 2,*

        suma += 2

        n = n / 2

*# n must be odd at this point*

*# so a skip of 2 ( i = i + 2) can be used*

**for** i **in** range(3,int(math.sqrt(n))+1,2):

*# while i divides n , print i ad divide n*

**while** n % i== 0:

*#print i,*

            suma += i

            n = n / i

*# Condition if n is a prime*

*# number greater than 2*

**if** n > 2:

*#print n*

        suma += n

**return** suma

**def** EsPrimo(n):

**if** (n < 2): **return** False

**if** (n == 2): **return** True

**if** (n % 2 == 0): **return** False

*#for(int i=3; i\*i<=n; i+=2)*

    i = 3

**while**(i \* i <= n):

**if** (n % i == 0): **return** False

        i += 2

**return** True

**def** mult\_primefactor\_sum( a,  b) : *# [a, b] range of numbers included a and b*

    ans = []

*#for (int i = a; i <= b; i++)*

**for** i **in** range(a, b + 1):

**if** (EsPrimo(i) == False):

            suma = primeFactors(i)

**if** (i % suma == 0):

                ans.append(i);

**return** ans

**print**(mult\_primefactor\_sum(10, 100))