**5 kyu**

**When The Sum of The Divisors Is A Multiple Of The Prime Factors Sum**

8080% of 1526 of38[raulbc777](https://www.codewars.com/users/raulbc777)

Python

* [TRAIN AGAIN](https://www.codewars.com/kata/when-the-sum-of-the-divisors-is-a-multiple-of-the-prime-factors-sum/train/python)
* [NEXT KATA](https://www.codewars.com/trainer/python)

Details

[Solutions](https://www.codewars.com/kata/when-the-sum-of-the-divisors-is-a-multiple-of-the-prime-factors-sum/solutions/python)

[Forks (4)](https://www.codewars.com/kata/when-the-sum-of-the-divisors-is-a-multiple-of-the-prime-factors-sum/forks/python)

[Discourse (3)](https://www.codewars.com/kata/when-the-sum-of-the-divisors-is-a-multiple-of-the-prime-factors-sum/discuss/python)

* Add to Collection
* |
* Share this kata:

The numbers 12, 63 and 119 have something in common related with their divisors and their prime factors, let's see it.

Numbers PrimeFactorsSum(pfs) DivisorsSum(ds) Is ds divisible by pfs

12 2 + 2 + 3 = 7 1 + 2 + 3 + 4 + 6 + 12 = 28 28 / 7 = 4, Yes

63 3 + 3 + 7 = 13 1 + 3 + 7 + 9 + 21 + 63 = 104 104 / 13 = 8, Yes

119 7 + 17 = 24 1 + 7 + 17 + 119 = 144 144 / 24 = 6, Yes

There is an obvius property you can see: the sum of the divisors of a number is divisible by the sum of its prime factors.

We need the function ds\_multof\_pfs() that receives two arguments: nMin and nMax, as a lower and upper limit (inclusives), respectively, and outputs a sorted list with the numbers that fulfill the property described above.

We represent the features of the described function:

ds\_multof\_pfs(nMin, nMax) -----> [n1, n2, ....., nl] # nMin ≤ n1 < n2 < ..< nl ≤ nMax

Let's see some cases:

ds\_multof\_pfs(10, 100) == [12, 15, 35, 42, 60, 63, 66, 68, 84, 90, 95]

ds\_multof\_pfs(20, 120) == [35, 42, 60, 63, 66, 68, 84, 90, 95, 110, 114, 119]

Enjoy it!!

<https://www.codewars.com/kata/when-the-sum-of-the-divisors-is-a-multiple-of-the-prime-factors-sum/python>

**import** math

**def** SumFactoresPrimos(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**  SumDivisores( n):

    div = []

    suma = 0

*#for (int i = 1; i \* i <= n; i++)*

    i = 1

**while**(i\*i <= n):

**if** (n % i == 0):

            suma += i;

**if** (n / i != i):

*#div.Add(n / i);*

                suma += (n / i)

        i+=1

**return** suma

**def** ds\_multof\_pfs(nMin, nMax):

    ans = []

*#for(int i =nMin; i<=nMax; i++)*

**for** i **in** range(nMin, nMax+1):

        sumDiv = SumDivisores(i)

        sumFP = SumFactoresPrimos(i)

**if**(sumDiv % sumFP ==0):

            ans.append(i)

**return** ans

**print**(ds\_multof\_pfs(10, 100))