Triste jour, Denise Vella-Chemla, juin 2026.

Aujourd'hui, j'ai reprogrammé la fonction d'Euler pour la somme des diviseurs. Le programme est

```
import math
from math import sqrt, floor
import numpy as np

def sommediv(n):
    somme = 0
    grandK = floor((1+sqrt(24*n+1))/6)
    for k in range(1,grandK+1):
        somme = somme+((-1)**(k-1))*(sommetilde(n,((3*k*k-k)/2))
                                    +sommetilde(n,((3*k*k+k)/2)))
    return(somme)

def sommetilde(x,y):
    if x == y:
        return x
    else:
        if x > y:
            return(tabsommediv[int(x-y)])
        else:
            return(0)

tabsommediv = np.zeros(105,dtype = 'int')
tabsommediv[1] = 1
for n in range(101):
    tabsommediv[n] = sommediv(n)
    print(n, ' --> ',tabsommediv[n])
```

J'ai pensé "Et si à la place des pentagonaux, je mettais des hexagonaux, qu'est c'que j'risque, j'le tente...", comme ça :

```
import math
from math import sqrt, floor
import numpy as np

def sommehasard(n):
    somme = 0
    grandK = floor((1+sqrt(24*n+1))/6)
    for k in range(1,grandK+1):
        somme = somme+((-1)**(k-1))*(sommetilde(n,k*(2*k-1))
                                    +sommetilde(n,(k*(2*k+1))))
    return(somme)

def sommehasardtilde(x,y):
    if x == y:
        return x
    else:
        if x > y:
            return(tabsommediv[int(x-y)])
```

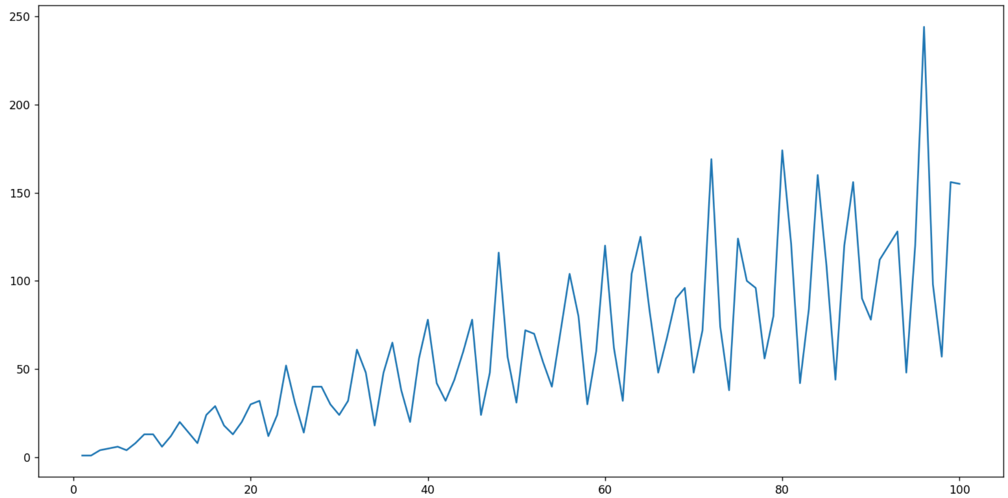
```
        else:
            return(0)

tabsommediv = np.zeros(105,dtype = 'int')
tabsommediv[1] = 1
for n in range(101):
    tabsommediv[n] = sommehasard(n)
    print(n,' —> ',tabsommehasard[n])
```

J'ai obtenu ça :

h(1) = 1	h(21) = 32	h(41) = 42	h(61) = 62	h(81)=121
h(2) = 1	h(22) = 12	h(42) = 32	h(62) = 32	h(82) = 42
h(3) = 4	h(23) = 24	h(43) = 44	h(63)=104	h(83) =84
h(4) = 5	h(24) = 52	h(44) = 60	h(64)=125	h(84)=160
h(5) = 6	h(25) = 31	h(45) = 78	h(65) = 84	h(85)=108
h(6) = 4	h(26) = 14	h(46) = 24	h(66) = 48	h(86) = 44
h(7) = 8	h(27) = 40	h(47) = 48	h(67) = 68	h(87)=120
h(8) = 13	h(28) = 40	h(48)=116	h(68) =90	h(88)=156
h(9) = 13	h(29) = 30	h(49) = 57	h(69) = 96	h(89) = 90
h(10) = 6	h(30) = 24	h(50) = 31	h(70) = 48	h(90) = 78
h(11) = 12	h(31) = 32	h(51) = 72	h(71) = 72	h(91)=112
h(12) = 20	h(32) = 61	h(52) = 70	h(72)=169	h(92)=120
h(13) = 14	h(33) = 48	h(53) = 54	h(73) = 74	h(93)=128
h(14) = 8	h(34) = 18	h(54) = 40	h(74) = 38	h(94) = 48
h(15) = 24	h(35) = 48	h(55) = 72	h(75)=124	h(95)=120
h(16) = 29	h(36) = 65	h(56)=104	h(76)=100	h(96)=244
h(17) = 18	h(37) = 38	h(57) = 80	h(77) = 96	h(97) = 98
h(18) = 13	h(38) = 20	h(58) = 30	h(78) = 56	h(98) = 57
h(19) = 20	h(39) = 56	h(59) = 60	h(79) = 80	h(99)=156
h(20) = 30	h(40) = 78	h(60)=120	h(80) =174	h(100)=155

J'ai cherché dans l'OEIS : je suis donc tombée, au culot, sur la séquence A113184 qui s'intitule "*Différence absolue entre la somme des diviseurs impairs et la somme des diviseurs pairs de n*". Fou, non ?!
Le graphique de cette fonction est :



Avec une petite modification, on obtient une fonction bien plus resserrée mais il ne semble pas qu'elle soit périodique, même si elle semble tourner tout le temps sur les mêmes nombres.

```
def sommehasard(n):
somme = 0
grandK = floor((1+sqrt(24*n+1))/6)
for k in range(1,grandK+1):
    somme = somme+((-1)**(k-1))*(sommehasardtilde(n,k*(k-1))
                                +sommehasardtilde(n,k*(k+1)))
return(somme)
```