

```

# Problem119_2_commenté.py

01| from math import floor, log, exp, ceil
02| import time
03|
04| def sommeChiffreNombre(nombre):
05|     '''
06|     :param nombre: nombre à décomposer
07|     :return: somme des chiffres de nombre
08|     '''
09|     return sum([ int(c) for c in str(nombre) ])      # transforme le nombre en chaîne de caractères pour en faire la somme
10|                                                     # de ses caractères
11|
12|
13| def A(n, vMax):
14|     '''
15|     :param n: terme dont on cherche la valeur
16|     :param vMax: variable permettant de borner a et b
17|     :return: n-ème terme répondant aux critères du problème
18|     '''
19|     listeNombre = []
20|     bMaximum = 2
21|     aMaximum = 2
22|     listebMaximum = {}
23|     while len(listeNombre) < n:                      # tant qu'on a pas trouvé le nombre que l'on recherche
24|         '''
25|         aPrevious, aMaximum, bMinimum, bMaximum et listebMaximum sont des outils nous permettant d'incrémenter vMax sans avoir à tout re-tester. En effet lorsque
26|         vMax est changé il faut retester sur les anciens a, les nouvelles puissances. Cependant sur les nouveaux a il faut tester toutes les puissances possibles. C'est
27|         pourquoi nous avons besoin de ces outils. Il n'est donc pas nécessaire ici d'expliquer en détail comment ils sont manipulés. Ainsi, nous ne détaillerons pas les
28|         opérations impliquant ces variables.
29|         '''
30|         a = 2
31|         aPrevious = aMaximum
32|         aMaximum = exp(log(vMax)/2)                  # on borne a grâce à la valeur minimum que b peut prendre (2) et grâce à vMax
33|         while a < aMaximum:                          # et on teste ainsi tous les a possibles
34|             if a >= aPrevious:
35|                 bMinimum = 2
36|             else:
37|                 bMinimum = listebMaximum[a]
38|                 bMaximum = floor(log(vMax) / log(a))+1 # on borne b (les puissances) en fonction de a
39|                 listebMaximum[a] = bMaximum
40|                 for b in range(bMinimum, bMaximum):  # ainsi pour chaque a on fait varier les puissances
41|                     value = a**b                     # et on stocke dans value
42|                     if sommeChiffreNombre(value) == a: # Si a est la somme des chiffres de value alors value est un des nombres qui nous intéresse
43|                         listeNombre.append(value)     # on l'ajoute donc à notre liste
44|                     a += 1
45|                 vMax *= 5**ceil((n - len(listeNombre))/2) # on augmente vMax en fonction de la taille de la liste
46|             listeNombre.sort()                        # on trie la liste afin d'obtenir le bon terme recherché
47|         return listeNombre[n-1]
48|
49| nbTeste = int(input("Rentrez le n-ème terme que vous souhaitez obtenir : "))
50| vTeste = int(input("Rentrez votre 'vMax' : "))
51| begin = time.time()
52| print("Résultat au problème 119 avec", nbTeste, "comme entrée :", A(nbTeste, vTeste))
53| print(f"Duration = {time.time() - begin} seconds to complete.")

```