

## nphermier\_4a

```
1  #!/usr/bin/env python3.6
2  # -*- coding: utf-8 -*-
3  # Série première. En Général Python
4  # MusicAtoumic nphermier_4a Le 16 septembre 2017
5
6  import time
7  import copy
8
9  t = time.time()
10
11 nombre = 8888999944443333
12 typ6 = nombre % 6
13 carre = int(nombre ** .5)
14 hautniveau = []
15 horscourse = []
16
17 if typ6 in (1, 5):
18     cartyp6 = int(nombre ** .473)
19     borne = int(nombre ** .4)
20 else:
21     cartyp6 = int(nombre ** .437)
22     borne = int(nombre ** .36)
23
24 """Pontage niveau haut:
25 Définir le carré en deux lignes (1&5)
26 Lié au reste de la division :carre % 6:""
27 carrel = carre
28 if carre % 6 == 1:
29     carre5 = carrel + 4
30 else:
31     differe = (carre % 6) - 1
32     carrel -= differe
33     carre5 = carrel + 4
34
35 """Bas niveau des premiers:
36 Création du haut niveau des nombres premiers
37 Lié au bas niveau des nombres premiers"""
38 basniveau = [2, 3, 5]
39 for i in range(7):
40     if i in basniveau and not nombre % i:
41         hautniveau.append(i)
42
43
44 def compare(c):
45     for haut in hautniveau:
46         if not c % haut:
47             horscourse.append(c)
48             break
49     else:
50         hautniveau.append(c)
51         hautniveau.sort()
52
53
54 """Hauts niveaux premiers:
55 Chercher dans l'alignement des nombres premiers"""
56 for p1 in range(7, cartyp6, 6):
57     p5 = p1 + 4
58     # Lorsque le haut niveau n'a pas d'élément bas
```

```

59     if not hautniveau:
60         if not nombre % p1:
61             hautniveau.append(p1)
62         if not nombre % p5:
63             hautniveau.append(p5)
64         # Partie inverse
65         if not nombre % carre1:
66             hautniveau.append(carre1)
67         if not nombre % carre5:
68             hautniveau.append(carre5)
69     # Lorsque le haut niveau a un élément bas
70     else:
71         if not nombre % p1 and p1 not in hautniveau:
72             compare(p1)
73         if not nombre % p5 and p5 not in hautniveau:
74             compare(p5)
75         # Partie inverse
76         if not nombre % carre1 and carre1 not in hautniveau:
77             compare(carre1)
78         if not nombre % carre5 and carre5 not in hautniveau:
79             compare(carre5)
80     carre1 -= 6
81     carre5 -= 6
82     if p1 > borne and 1 < len(hautniveau):
83         produit = 1
84         hautniveau.reverse()
85         for unit in hautniveau:
86             produit *= unit
87             result = nombre // produit
88             if result > 1:
89                 if produit > carre and result not in horscourse:
90                     compare(result)
91                     break
92                 elif produit not in horscourse:
93                     horscourse.append(produit)
94                 elif nombre in horscourse:
95                     break
96         else:
97             # Dossier ('vide', horscourse)
98             break
99
100     """Désigne la communauté:
101     Les nombres premiers associés :hautniveau:
102     Produisent la communauté des multiples communs"""
103     hautniveau.append(1)
104     hautniveau.sort()
105     print('Premiers', hautniveau)
106     communs = copy.copy(hautniveau)
107     for com1 in hautniveau:
108         for com2 in communs:
109             produit = com1 * com2
110             # Condition de recherche
111             if produit > carre:
112                 break
113             elif produit not in communs and not nombre % produit:
114                 communs.append(produit)
115                 communs.sort()
116             # Ordonnance des communs
117
118     for i in communs:

```

```
119     print('{} * {} typ {}*{}'.format(i, nombre // i, i % 6, (nombre //
i) % 6))
120
121 print('nphermier_4a.py En:', time.time() - t)
122 print('', )
123
```