

nphermier_9a.py

```
1  #!/usr/bin/env python3.6
2  # -*- coding: utf-8 -*-
3  # Série première. En Général Python
4  # MusicAtoumic nphermier_9a Le 8 octobre 2017
5
6  import time
7  import copy
8
9  t = time.time()
10
11 nombre = 1234567890987654321
12 typ6 = nombre % 6
13 carre = int(nombre ** .5)
14 hautniveau = []
15 horscourse = []
16 nomproduit = 0
17
18 """Borne & cartyp6
19 Borne : Limite en pleine forêt
20 Cartyp6 : Limite en plein ciel"""
21 cartyp6 = int(nombre ** .36)
22 borne = int(nombre ** .33)
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     if c not in hautniveau:
46         if c not in horscourse:
47             horscourse.append(c)
48         for haut in hautniveau:
49             if not c % haut:
50                 break
51     else:
52         if c <= carre:
53             hautniveau.append(c)
54
55
56 """Hauts niveaux premiers:
57 Chercher dans l'alignement des nombres premiers"""
58 for p1 in range(7, cartyp6, 6):
```

```

59     p5 = p1 + 4
60     # Lorsque le haut niveau n'a pas d'élément bas
61     if not hautniveau:
62         if not nombre % p1:
63             hautniveau.append(p1)
64         if not nombre % p5:
65             hautniveau.append(p5)
66         # Partie inverse
67         if not nombre % carre1:
68             hautniveau.append(carre1)
69         if not nombre % carre5:
70             hautniveau.append(carre5)
71     # Lorsque le haut niveau a un élément bas
72     else:
73         if not nombre % p1 and p1 not in hautniveau:
74             compare(p1)
75         if not nombre % p5 and p5 not in hautniveau:
76             compare(p5)
77         # Partie inverse
78         if not nombre % carre1 and carre1 not in hautniveau:
79             compare(carre1)
80         if not nombre % carre5 and carre5 not in hautniveau:
81             compare(carre5)
82     carre1 -= 6
83     carre5 -= 6
84
85     """Condition de la borne:
86     Nombres premiers complémentaires"""
87     if p1 > borne and 1 < len(hautniveau):
88         produit = 1
89         # Production des termes (1&5)
90         # Résulte :result: de type (%6) variable
91         for unit in hautniveau:
92             produit *= unit
93             result = nombre // produit
94             if produit < nombre:
95                 if produit > carre:
96                     compare(result)
97                 else:
98                     compare(produit)
99             elif produit == nombre:
100                 for h in horscourse:
101                     if h not in hautniveau:
102                         nomproduit = 0
103                         break
104                 else:
105                     # Condition déterminée
106                     nomproduit = 1
107         else:
108             """La production terminée
109             Le hors course de son élément supérieur :max():
110             Le haut niveau de son élément inférieur :min():
111             """
112             while 1:
113                 fx_ip = nombre // max(horscourse)
114                 fx_mu = fx_ip * min(hautniveau)
115                 if not nombre % fx_mu:
116                     # Condition de la partie supérieure
117                     if fx_mu > carre:
118                         # Dépassement supérieur
119                         while 1:

```

```

120         fx_in = nombre // fx_mu
121         if fx_in % 6 in (1, 5) and not nombre %
fx_in:
122             compare(fx_in)
123             break
124         elif not nombre % fx_in:
125             fx_mu *= min(hautniveau)
126         else:
127             break
128             break
129         else:
130             break
131         break
132
133
134 """Désigne la communauté:
135 Les nombres premiers associés :hautniveau:
136 Produisent la communauté des multiples communs"""
137 hautniveau.append(1)
138 hautniveau.sort()
139 print('Premiers', hautniveau)
140 communs = copy.copy(hautniveau)
141 if nomproduit == 0:
142     for com1 in hautniveau:
143         for com2 in communs:
144             produit = com1 * com2
145             # Condition de recherche
146             if produit > carre:
147                 break
148             elif produit not in communs and not nombre % produit:
149                 communs.append(produit)
150                 communs.sort()
151                 # Ordonnance des communs
152
153 for i in communs:
154     print('{} * {} typ {}*{}'.format(i, nombre // i, i % 6, (nombre //
i) % 6))
155
156 print('nphermier_9a.py En:', time.time() - t)
157 print('', )
158

```