

## nphullmen\_6a.py

```
1  #!/usr/bin/env python3.6
2  # -*- coding: utf-8 -*-
3  # Série première. En Général Python
4  # MusicAtoumic nphullmen_6a Ce 11 décembre 2017
5  """Python Compulsion Commune"""
6
7  import time
8
9  time_ = time.time()
10
11
12  def zero(z):
13      if z == 1:
14          z = 0
15          print('Premiers (', z, ')ex\n', '[0]')
16          print('{} * {} Types {}&{}'
17                .format(z, z, z % 6, z % 6))
18          print('nphullmen_6x.py En: \n', time.time() - time_)
19      else:
20          print('nphullmen_6x.py En: \n', time.time() - time_)
21
22
23  # -----
24  # :nombre: objet unique
25  nombre = 1285660625959
26  # :breakComm: printage communs
27  breakComm = 1
28  # :zero: instruction zéro
29  if nombre == 0:
30      zero(breakComm)
31  # :carre: service axial
32  carre = int(nombre ** .5)
33
34  hautniveau = [] # Liste premiers
35  horscourse = [] # Liste communs ip&dv
36  hautmulti_ = [1] # Mul.hautniveau
37  horsmulti_ = [1] # Mul.horscourse
38  cl2tableau = [] # Liste communs dv
39
40  # Lecteurs initiaux
41  p1 = [7]
42  p5 = [11]
43  q1 = [0]
44  q5 = [0]
45
46  """Borne & cartyp6
47  Borne : Limite en pleine forêt
48  Cartyp6 : Limite en plein ciel"""
49  cartyp6 = [carre]
50  borne = int(cartyp6[0] ** .999)
51  borne = [borne]
52
53
54  def formel15(r1_, rng_, r6_):
55      """Positionnement des Lecteurs (bas/hauts)
56          P1: Point (+6) Type (1)
57          Position inférieure :ip: Inf.carre
58          Q1: Point (-6) Type (5)
```



```

120                                     c12tableau.sort()
121 if c12tableau:
122     for ctab in c12tableau:
123         iptab = nombre // ctab
124         # :ctab: Occurence :c12:
125         if iptab not in horscourse:
126             for hh in hautniveau[1:]:
127                 if not iptab % hh:
128                     break
129             else:
130                 compare(iptab)
131     if hautmulti_[0]:
132         c12ip = nombre // max(hautniveau)
133         # :c12ip: Sup.carre(dv) de Max.premier(ip)
134         if max(c12tableau) > c12ip:
135             hautmulti_[0] = 0
136
137 if hautmulti_[0]:
138     horsmulti_[0] = 1
139     # Production hors course
140     for hho in horscourse:
141         if horsmulti_[0] <= nombre:
142             horsmulti_[0] *= hho
143         cartyp6[0] = int((nombre // max(hautniveau)) ** .499)
144         brn = int(cartyp6[0] ** .5)
145         borne[0] = brn
146
147     # Terme commun
148     if max(horscourse) <= carre:
149         if hautmulti_[0]:
150             hdv = hautmulti_[0] * hautniveau[1]
151             # :hdv: Son.carre(dv) par Min.hautniveau
152             if hdv >= carre and not nombre % hdv:
153                 hip = nombre // hdv
154                 # :hip: Son(ip) du Son.carre(dv)
155                 if hip not in horscourse:
156                     compare(hip)
157
158     if hautmulti_[0]:
159         # Termes racine
160         if hautmulti_[0] == nombre:
161             hautmulti_[0] = 0
162         elif hautmulti_[0] * hautniveau[1] >= carre \
163             and hautmulti_[0] == max(hautniveau):
164             hautmulti_[0] = 0
165         elif hautmulti_[0] * hautniveau[1] >= nombre:
166             hautmulti_[0] = 0
167         elif max(horscourse) == nombre:
168             hautmulti_[0] = 0
169         else:
170             # Différentiel comparatif
171             dvinf = nombre // max(horscourse)
172             ipsup = dvinf // max(horscourse)
173             if len(hautniveau) > 2:
174                 if ipsup >= carre:
175                     rng = int(
176                         (nombre // max(hautniveau)) ** .4534)
177                     r6 = rng % 6
178                     formel5(0, rng, r6)
179                 else:
180                     sqinf = int(dvinf ** .5)

```

```

181         if not nombre % sqinf \
182             and sqinf % 6 in(1, 5):
183             compare(sqinf)
184         else:
185             sqinf = int(dvinf **.4)
186             for e in range(2, sqinf):
187                 if not dvinf % e:
188                     if e % 6 in (1, 5) \
189                         and e not in
horscourse:
190                             compare(e)
191                             break
192             else:
193                 hautmulti_[0] = 0
194
195
196 """Bas niveau des premiers:
197 Création du haut niveau des nombres premiers
198 Lié au bas niveau des nombres premiers
199 Composition des hors course"""
200 basniveau = [1, 2, 3, 5]
201 for i in range(1, 7):
202     if not nombre % i and i <= carre:
203         def rang(ti, i_):
204             if ti == 1: # Nombre Premier(basniveau)
205                 hautniveau.append(i_)
206                 horscourse.append(i_)
207             else: # Nombre Commun(basniveau)
208                 horscourse.append(i_)
209
210
211         if i in basniveau:
212             rang(1, i)
213         else:
214             rang(2, i)
215 if horscourse:
216     for h in horscourse:
217         for o in horscourse:
218             oh = h * o
219             if not nombre % oh and oh not in horscourse:
220                 compare(oh)
221
222 """Hauts niveaux premiers: cartyp6
223 Itérer dans l'alignement des nombres premiers
224 Condition évolutive des lecteurs"""
225 while p1[0] <= cartyp6[0]:
226
227     if hautmulti_[0] == 0:
228         break
229     elif p1[0] > q1[0]:
230         break
231     elif p1[0] >= cartyp6[0]:
232         break
233     elif p1[0] >= int(borne[0] **.5):
234         if hautmulti_[0] != 0:
235             dvhors = nombre // max(horscourse)
236             sqhors = int(dvhors **.5)
237             for ds in range(2, int(sqhors **.5)):
238                 if not dvhors % ds and ds not in horscourse:
239                     compare(ds)
240         else:

```

```

241             break
242
243     # Lectures
244     if not nombre % p1[0] and p1[0] not in hautniveau \
245         and p1[0] not in horscourse:
246         compare(p1[0])
247     if not nombre % p5[0] and p5[0] not in hautniveau \
248         and p5[0] not in horscourse:
249         compare(p5[0])
250     if not nombre % q1[0] and q1[0] not in hautniveau \
251         and q1[0] not in horscourse:
252         compare(q1[0])
253     if not nombre % q5[0] and q5[0] not in hautniveau \
254         and q5[0] not in horscourse:
255         compare(q5[0])
256
257     p1[0] += 6
258     p5[0] += 6
259     q1[0] -= 6
260     q5[0] -= 6
261
262     if nombre != 0:
263         """Désigne la communauté:
264         Les nombres premiers associés :hautniveau:
265         Produisent la communauté des multiples communs"""
266         if max(horscourse) > carre:
267             hi = len(horscourse) // 2
268         else:
269             hi = len(horscourse)
270         hautniveau.sort()
271         print('Premiers (', hi, ')ex\n', hautniveau)
272
273         for i in sorted(horscourse):
274             if breakComm == 0:
275                 ha = horscourse[hi - 1]
276                 hu = horscourse[0]
277                 print('Duos (inf)*(sup)|(sup)*(inf) \n {} * {} Types
278 {}&{}'
279                     .format(hu, nombre // hu, hu % 6, (nombre // hu) %
280 6))
281                 print(' {} * {} Types {}&{}'
282                     .format(ha, nombre // ha, ha % 6, (nombre // ha) %
283 6))
284                 break
285             else:
286                 print('{} * {} Types {}&{}'
287                     .format(i, nombre // i, i % 6, (nombre // i) % 6))
288
289     print('nphullmen_6a.py En: \n', time.time() - time_)
290     print('', )

```