

# ORGANISATION D'une énumération des nombres superposés  
# Descriptions réalisées en connaissances humaines  
# Quand, le langage du python exécute une série logique  
# Ce jour, jeudi 8 mai 2014, à environ seize heures

Bonjour, enfin du code numérique !

Mine de rien ce programme (**exercicepy3**), calcule les 4/6ème de tous les entiers. La forme syntaxique du langage « python » et l'algorithme du modèle **modulo**, vont à tous les deux calculer la signature des nombres entiers.

Comment ça marche :

- Par définition, les nombres entiers dans leurs généralités, rassemblent une quantité de types. Ces suivants, selon les chiffres :

- o 1 : PREMIER MULTIPLE 1 \* IMPAIR
- o 2 : PREMIER MULTIPLE 2 \* PAIR
- o 3 : PREMIER MULTIPLE 3 \* IMPAIR
- o 4 : MULTIPLE DE 2 \* PAIR
- o 5 : PREMIER MULTIPLE 5 \* IMPAIR
- o 6 : MULTIPLE DE 3 \* PAIR

- L'opération sur un entier ici, prend son calcul par la subdivision du nombre. Le transforme pour lui donner sa valeur réelle :

- o La valeur des nombres est, qu'ils ont tous un unique point commun. C'est avec le reste de la division, que le chiffre 6 développe ses relatives numériques. Le module mathématique qui restitue le reste de la division et son caractère spécial « % ».

Soit, lorsqu'on calcule le rapport «  $C = 3/6$  » ; C en est le quotient, 3 le dividende, 4 le diviseur...

- o Alors, «  $D = 3\%6$  » est le reste de la division («  $C = 3/6$  »). Le module modulo, utilise une constante qui s'avère être parfaite à savoir l'aide du calcul, qui a chaque nombre entier est assuré de produire son type qualitatif (pair/impair...etc)

- Le programme est écrit en langage **python**, licence **Open Source** d'actualité comme **blender** a signé toute la puissance de ce code :

- o Le code « **exercicepy3** » est principalement logique, sans lui, l'algorithme procéderait à d'autres expressions du langage, ce qui ne serait pas utile. Seulement, si le fait d'un résultat nécessite la construction d'une procédure complexe. Ce code est fondamentalement axé sur un tri séquentiel. Lorsque l'exercice est exécuté, il effectue sa propre et première opération sur une zone limitée aux six chiffres du panel hexanumérique.

o Le panel hexanumérique produit six rangées d'entiers consécutifs, il aligne quantitativement et qualitativement. Il est également confronté à la possibilité de compter deux séries d'entiers impairs pouvant intégrer la fonction de : Premier Multiple commun. Le nombre entier premier conduit à la multiplicité, en évoluant comme un tempérament.

• La représentation entière numérise les nombres entiers, cette synthétisation supporte quelques qualités supplémentaires :

o Dans l'intervalle des entiers « Premier Multiple commun », il y a les premiers nombres premiers. Le tout premier chiffre à valeur entière c'est 1. Ce nombre (1) est le premier multiple universel, où tous les nombres entiers sont multiples de 1. Ce chiffre universel est un « Premier multiple fondamental », à moindre valeur multiple le Premier Multiple commun a un intervalle plus élargi. Les premiers multiples communs (sont : '2 3 5').

o L'intervalle relatif au nombre premier fondamental, est relatif à lui-même. Il n'y a aucun espace de libre, aucun intervalle dans la séparation de ces deux mêmes unités. A l'image d'un collier de perles, représentant la répétition d'une seule entité. En élargissant le pas numérique, en passant au deuxième chiffre de la série des nombres entiers. Soit (2), ce nombre a une parité dans la première vague hexanumérique. Il y a dans ses rangs un nombre isolé, qu'en étant uniquement Premier Multiple Pair d'un Premier Multiple Commun...

### § PREMIER MULTIPLE FONDAMENTAL

- Comme déjà dit : Le seul nombre à être intégré à tous les nombres.
- Le chiffre 1.

### § PREMIER MULTIPLE COMMUN

- Composition du panel hexanumérique, dans une forme secondaire.
- Le chiffre 2.
- le chiffre 3.
- le chiffre 5.

### § PREMIER MULTIPLE PAIR

- Composition au même panel, un nombre multiple dans la chambre des premières relations.
- Le chiffre 4.

### § PREMIER MULTIPLE IMPAIR

- Le chiffre concerné représente une limite extrême, les nombres supérieurs forment des couchettes.
- Le chiffre 6.

\*\*\*

```
# Définition exacte des nombres
# Dans le système hexadécimale
```

```
for n in range(1,100):
```

```
    if (n <= 6) :
```

```
        tour = 0
```

```
    else :
```

```
        tour = 1
```

```
x = n%6    # Module d'analyse
```

```
if (x == 1) :
```

```
    if (tour == 0) :
```

```
        print (n, 'premier multiple')
```

```
    else :
```

```
        print (n, 'possible premier')
```

```
if (x == 2) :
```

```
    if (tour == 0) :
```

```
        print (n, 'premier multiple')
```

```
    else :
```

```
        print (n, 'multiple de 2')
```

```
if (x == 3) :
```

```
    if (tour == 0) :
```

```
        print (n, 'premier multiple')
```

```
    else :
```

```
        print (n, 'multiple de 3  
impair')
```

```
if (x == 4) :
```

```
    print (n, 'multiple de 2')
```

```
if (x == 5) :
```

```
    if (tour == 0) :
```

```
        print (n, 'premier multiple')
```

```
    else :
```

```
        print (n, 'possible premier')
```

```
if (x == 0) :
```

```
    print (n, 'multiple de 3 pair')
```