TP Informatique commune 1A : Fonctions, listes

Exercice 1

Essayer d'exécuter le code ci-dessous: que se passe-t-il ?

```
def estbis(a):
  if (a%4==0 and a%100!=0) or (a%400==0):
    r=True
  else:
    r=False
  return r
```

Taper ensuite dans la console `estbis(2000)`, `estbis(2002)`, `estbis(2004)`, `estbis(2100)`.

Comparer avec l'exercice 1 du TP3, donner les avantages et inconvénients des deux codes.

Exercice 2 : Nombres premiers

1. Écrire une fonction `petit_div` associant à tout entier naturel N (supérieur ou égal à 2) son plus petit diviseur (supérieur ou égal à 2).

2. Écrire une fonction `estpremier` associant à tout entier naturel N la valeur `True` si N est premier et `False` sinon. Compléter :

`N`	`estpremier(N)`
0	
1	
2	
9	
11	
169	
1001	
11111111111111111	

3. Écrire une fonction `effectif` déterminant le nombre de nombres premiers entre a et b (compris).

   On vérifiera que `effectif(0,10)` donne bien 4.

   Compléter :

`N`	0	10	10^2	10^3	10^4	10^5
`effectif(N,N+1000)`						

4. Écrire une fonction `jumeau(N)` renvoyant un couple de nombre premiers jumeaux (i.e. dont la différence vaut 2) tous deux strictement supérieurs à N. Que renvoie jumeau(10^6)?

   On ne sait pas s'il existe une infinité de nombres premiers jumeaux.

5. Écrire une fonction `gap(N)` qui renvoie un entier A tel qu'il n'y ait aucun nombre premier entre A et $A+N$ (compris). Que renvoie `gap(30)`?

Exercice 3 : Manipulation de listes

1. Écrire une suite d'instructions permettant de construire la liste des nombres de la forme k^2+1 où k est un entier positif, et tels que $k^2+1 \leqslant 120$.

2. Modifier le programme précédent pour ne garder que les nombres premiers.

3. Écrire une fonction `liste_div(N)` prenant en entrée un entier supérieur ou égal à 2 N et qui renvoie la liste de ses diviseurs. En déduire une autre version de la fonction qui teste si N est premier.

4. Écrire une fonction `lesJumeaux(N)` prenant en entrée un entier positif N et qui renvoie la liste de tous les couples de nombres premiers jumeaux (p,q) tels que $q \leqslant N$. Combien y'a-t-il de tels couples pour $N = 1000$?

5. Écrire une fonction `moitie(L)` prenant en entrée une liste L de nombres et qui renvoie une autre liste M construite à partir de L en ne gardant qu'un terme sur deux (ceux d'indice pair). Par exemple `moitie([1,2,3,4,5,6]` devra renvoyer `[1,3,5]`. (Rappel : on numérote les indices à partir de 0 !)

Exercice 4 : Suite de Syracuse, le retour !

On rappelle la définition de la suite de Syracuse u : $u_0 \in \mathbb{N}^\star$ et

$$\forall n \in \mathbb{N}, \quad u_{n+1} = \begin{cases} 3u_n + 1 & \text{si } u_n \text{ est impair} \\ \frac{u_n}{2} & \text{si } u_n \text{ est pair} \end{cases}$$

Pour un entier u_0 donné, on nomme:

- *durée du vol*, l'entier défini par : $\min\{N \in \mathbb{N}^\star, u_N = 1\}$
- *altitude maximale du vol*, l'entier défini par : $\max\{u_n, n \in \mathbb{N}\}$
- *durée de vol en altitude*, l'entier défini par : $\max\{n \in \mathbb{N}, \forall k \in [\![0,n]\!], u_k \geqslant u_0\}$

1. Écrire une fonction `syracuse(u0,n)` prenant comme argument u_0 et n et donnant la valeur de u_n.

2. Écrire dans chacun des cas suivants, une fonction prenant en entrée le premier terme u_0 et renvoyant:

   (a) la liste des N premiers termes de la suite (N étant également un paramètre d'entrée de la fonction);
   (b) le vol (c'est-à-dire la liste de tous les termes de la suite jusqu'à atteindre 1) ;
   (c) la durée du vol (soyez malins !);
   (d) l'altitude maximale du vol;
   (e) la durée de vol en altitude.

   Préférez-vous utiliser ces fonctions ou les programmes du TP3 ?