PYTHON : LES BOUCLES FOR ET WHILE

EXERCICE 1 Tester les algorithmes suivants, puis les reprogrammer avec des boucles `for`.

```
#1
i=1
while i<10:
  i+=1
  print(i)
print('fini')

#2
i=1
while i<10:
  print(i)
  i+=1
print('fini')
```

```
#3
i=0
while i<=10:
  print(i)
  i+=1
print('fini')

#4
i=0
while i<11:
  print(i)
  i+=2
print('fini')
```

EXERCICE 2 (marche arrière)

1) Programmer une boucle **while** qui écrit les nombres de 100 à 1 dans l'ordre décroissant.

2) Faire de même avec une boucle **for**.

Exercice type à savoir refaire rapidement :

EXERCICE 3 (suite définie par une relation de récurrence)
Soit la suite définie par $u_0 = 1$ et $\forall n \in \mathbb{N}$, $u_{n+1} = \sqrt{u_n + n}$

1) Justifier que la suite est bien définie pour tout $n \in \mathbb{N}$.

2) Écrire un programme informatique qui calcule le terme u_{100}.

3) Écrire une fonction Python **suite(n)** qui prend en argument l'entier naturel n et qui renvoie la valeur de u_n.
Vérifier que la fonction renvoie la même valeur pour u_{100} que le programme précédent et vérifier à la main les premiers termes de la suite.

4) Écrire une fonction Python **somme(n)** qui renvoie la valeur $\sum_{k=0}^{n} u_k$.

5) **(a)** Justifier que la suite u est croissante.
(b) Montrer que la suite u diverge vers $+\infty$.
(c) En déduire que $\forall M \in \mathbb{R}$, $\exists n \in \mathbb{N}$, tel que $u_n > M$.

6) Écrire une fonction **rangSup(M)** qui prend en argument un nombre M et renvoie le premier indice n tel que $u_n \geq M$.

7) Écrire une fonction **rangInf(M)** qui prend en argument un nombre M et renvoie le dernier indice n tel que $u_n < M$.

8) Écrire une fonction **termeSup(M)** qui prend en argument un nombre M et qui renvoie le premier terme u_n de la suite tel que $u_n \geq M$.

9) Écrire une fonction **termeInf(M)** qui prend en argument un nombre M et qui renvoie le dernier terme u_n de la suite tel que $u_n < M$.

EXERCICE 4 (logarithme)

1) Écrire une fonction **log2** qui prend en argument un entier n, avec $n \geq 1$ et calcule le plus grand entier k tel que $2^k \leq n$.

2) Vérifier la fonction avec l'entier 128.

3) Vérifier la fonction avec l'entier 9876543210123456789.

EXERCICE 5 (Suite de Syracuse)

Une suite de Syracuse est définie par

$$u_0 \in \mathbb{N}^*, \quad \text{et} \quad \forall n \in \mathbb{N}^*,\ u_{n+1} = \begin{cases} \frac{u_n}{2} & \text{si } u_n \text{ est pair} \\ 3u_n + 1 & \text{si } u_n \text{ est impair} \end{cases}$$

Dans la suite, on considérera que la conjecture de Syracuse est juste : pour toute condition initiale $u_0 \in \mathbb{N}^*$, la suite atteint la valeur 1.

1) Programmer une fonction **syracuse(u0, n)** qui prend en argument un entier $u0 \geq 1$, et un entier $n \in \mathbb{N}$ et qui calcule le terme u_n de la suite de Syracuse.

2) Programmer une fonction **tempsVol(u0)** qui prend en argument un entier $u0 \geq 1$, et qui renvoie le premier entier n tel que $u_n = 1$.

3) Programmer une fonction **maximum(u0)** qui prend en argument un entier $u0 \geq 1$, et qui renvoie le maximum de la suite (u_n).

4) Programmer une fonction **tpsVolMax(n)** qui prend en argument un entier $n \geq 1$, et qui renvoie le plus petit entier $p \in [\![1, n]\!]$ pour lequel la suite de premier terme $u_0 = p$ a le plus grand temps de vol.

EXERCICE 6 (S'il reste du temps)

Nous chargeons le module qui permet de tracer des courbes en Python.

```
from matplotlib.pyplot import *

plot(3,5,'ro')    # trace un rond (o) rouge (r) de coordonnées (3,5)
plot(1,-3,'b.')   # trace un point (.) bleu (b) de coordonnées (1,-3)
```

Pour $n \in \mathbb{N}^*$, et en vous aidant des commandes précédentes, écrire un programme Python qui trace les points dont les abscisses sont les valeurs de $u_0 \in [\![1, n]\!]$ et les ordonnées le temps de vol de la suite de Syracuse de premier terme u_0.