Boucles while — TP5

Exercice 1 Petits rappels

On rappelle la syntaxe d'une boucle **`while`**

```
while condition :
    # Instructions dans la boucle
# Instructions en dehors de la boucle
```

Q1 Dans le code ci-dessous, de quel type d'objet Python est "`condition`" ?

Q2 Supposons avoir défini précédemment une fonction f. Que mettre à la place de "`condition`" si je cherche le plus petit entier n tel que $f(n) < M$?

Q3 Que se passe-t-il à l'exécution de la boucle suivante ? Proposer un résultat puis recopier la boucle dans Spyder ou Pyzo pour vérifier.

```
i=20
while i>10:
    print(i)
    i+=1
```

Remarque

Exercice 2 À partir d'un certain rang

On considère la suite (u_n) définie par récurrence par $u_0 = 0$ et

$$\forall n \geq 0,\ u_{n+1} = n(-1)^n + u_n \frac{n-5}{3n+2}$$

Q1 Écrire une fonction `suite` qui prend en argument en entier naturel `n` et renvoie la valeur de u_{n}. On vérifiera que `suite(100)` renvoie -75,5.

Q2 Écrire une fonction `audela`, utilisant la fonction `suite`, qui prend en argument une variable `seuil` et qui renvoie le plus petit rang $N \in \mathbb{N}$ tel que $u_N \geq$ `seuil`. On vérifiera que `audela(10)` renvoie la valeur 13.

Q3 Écrire une fonction `endessous`, utilisant la fonction `suite`, qui prend en argument une variable `seuil` et qui renvoie le plus petit rang $N \in \mathbb{N}$ tel que $u_N <$ `seuil`. On verifiera que `endessous(-10)` renvoie la valeur 14.

Q4 Si on s'intéresse à "l'efficacité" des fonctions, était-ce judicieux d'utiliser la fonction `suite` pour les fonctions `audela` et `endessous` ? Réécrire ces fonctions de manière plus efficace, et vérifier qu'on obtient bien le même résultat.

Exercice 3 Approximation de π

La formule de Brent-Salamin (des noms de deux mathématiciens des années 1970) permet d'obtenir rapidement une bonne approximation de π (elle fut utilisée en 1999 pour obtenir plus de 206 millions de décimales de π !).

Cette méthode consiste à définir $a_0 = 1$, $b_0 = \frac{1}{\sqrt{2}}$, $t_0 = \frac{1}{4}$ et $p_0 = 1$ puis pour tout $n \geq 0$,

$$t_{n+1} = t_n - p_n \left(\frac{a_n - b_n}{2}\right)^2 \ , \ a_{n+1} = \frac{a_n + b_n}{2} \ , \quad b_{n+1} = \sqrt{a_n b_n} \ , \qquad \text{et} \qquad p_{n+1} = 2p_n.$$

Un théorème affirme alors que, lorsque a_n et b_n sont "proches", la valeur

$$\frac{(a_n + b_n)^2}{4t_n}$$

est une "bonne" approximation de π.
Écrire une fonction `approx` prenant en argument un réel `epsilon` et renvoyant l'approximation de π obtenue par cette méthode lorsque l'on s'arrête dès que $|a_n - b_n| \leq$ `epsilon`.
Testez votre fonction, puis écrivez une fonction `approx_bis` pour savoir également combien d'itérations ont été nécessaires pour obtenir le résultat.

Exercice 4 Conjecture de Syracuse

Soit $f : \mathbb{N}^* \longrightarrow \mathbb{N}^*$ la fonction définie par

$$f(k) = \begin{cases} \frac{k}{2} & \text{si } k \text{ est pair} \\ 3k+1 & \text{si } k \text{ est impair} \end{cases}$$

On appelle *«suite de Syracuse d'un entier N»* la suite (u_n) définie par $u_0 = N$ et $u_{n+1} = f(u_n)$ pour tout $n \geq 0$.

Q1 Calculer à la main la suite de Syracuse de 3. Que se passe-t-il lorsque la suite atteint la valeur 1 ?

> **Remarque**
> La conjecture de Syracuse affirme que toutes les suites de Syracuse des entiers positifs atteignent la valeur 1 au bout d'un certain temps. Cette conjecture a été vérifiée pour tous les entiers naturels N inférieurs à 2^{62}, mais on ignore encore si elle est vraie.

Q2 Écrire une fonction `f` prenant en argument un entier `k` ≥ 1 et renvoyant f(`k`).

Q3 Écrire une fonction `syracuse` prenant en argument deux entiers `N` et `n` et renvoyant la valeur `syracuse(N,n)` $= u_n$ où (u_n) est la suite de Syracuse de `N`. On vérifiera que `syracuse(15,9)` renvoie la valeur 40.

Q4 Écrire une fonction `TempsVol` prenant en argument un entier `N` et renvoyant la plus petite valeur de n telle que le n-ème terme de la suite de Syracuse de l'entier N vaut 1 (on suppose que ce terme existe, c'est-à-dire que la conjecture est vérifiée). On vérifiera que `TempsVol(15)` renvoie la valeur 17.

Q5 Tracer les valeurs de `TempsVol(N)` en fonction de `N` pour $1 \leq$ `N` ≤ 1000.

> **Remarque**
> Pour tracer un graphique, on importera la bibliothèque `matplotlib.pyplot` en écrivant en amont du code :
>
> ```
> import matplotlib.pyplot as plt
> ```
>
> On pourra ensuite dessiner les points du graphique un par un en utilisant la commande `plt.plot(x,y,'bo')` pour tracer un point bleu aux coordonnées `(x,y)`.

Q6 Que se passe-t-il si on choisit $N < 0$? On pourra consulter la vidéo du youtuber "El Jj" sur le sujet : `https://www.youtube.com/watch?v=BP2G28694z8`.