

# Informatique - TP 4

## Utilisation des boucles `while`

M. Marmorat, M. Morel

12 novembre 2024

### Exercice 1 Relisons le cours

**Q1** On rappelle la syntaxe d'une boucle `while` :

```
1 while condition:
2     #bloc d'instructions
3 #instructions hors de la boucle while
```

Que se passe-t-il si la condition `condition` n'est pas vérifiée ?

**Q2** Que produit l'exécution du code suivant ?

```
1 n = 0
2 while n < 10:
3     n = 2*n + 1
4     print(n)
5 print("Fin")
```

**Q3** Quelles différences existent entre une boucle `while` et une boucle `for` ?

**Q4** Attention à ce que le critère d'arrêt d'une boucle `while` soit atteint ! Que se passe-t-il à l'exécution du script suivant ?

```
1 compteur = 20
2 while compteur > 10:
3     print(compteur)
4     compteur += 1
```

Pourquoi ?

**Remarque 1.** Pour arrêter une boucle infinie, faites un clic droit sur le shell et cliquez sur "Interrompre".

## Exercice 2 Retour sur la factorielle

**Q1** Implémenter la fonction `seuil_factorielle` vue en cours, qui prend en argument un nombre réel positif  $M$  et qui **renvoie** le plus petit nombre entier  $n$  tel que  $n! \geq M$ .

**Q2** Rappeler (en calculant par une méthode de votre choix, avec votre intelligence ou avec Python) les valeurs de  $n!$  pour  $n$  allant de 0 à 7.

**Q3** Vérifier le fonctionnement de votre fonction `seuil_factorielle` sur les nombres d'entrée  $M = 5$ ,  $M = 530$  et  $M = 1000$ . Anticipez le résultat avant d'appeler votre fonction !

## Exercice 3 A partir d'un certain rang

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

$$u_{n+1} = u_n + \sqrt{n}.$$

**Q1** Écrire une fonction `suite` qui prend en argument un entier  $n$  et qui renvoie le nombre  $u_n$  (on utilisera une boucle `for`; pour utiliser la fonction  $\sqrt{\cdot}$  on pourra l'importer grâce à l'instruction

```
1 from math import sqrt
```

que l'on placera au tout début du script).

**Q2** Écrire une fonction `seuil` qui prend en entrée un nombre  $M$  et qui renvoie le plus petit entier  $n$  tel que  $u_n \geq M$  (on utilisera la fonction `suite` de la question précédente. On vérifiera que `seuil(30)=13`).

**Q3** Si l'on s'intéresse à l'efficacité de la fonction `seuil`, était-il judicieux d'appeler la fonction `suite`? Pourquoi? Proposer une implémentation plus efficace et vérifier que les résultats sont les mêmes.

## Exercice 4 Approximation de $\sqrt{2}$ par la méthode de Héron

On définit la suite  $(u_n)$  par  $u_0 = 1$  et pour tout  $n \in \mathbb{N}$ ,

$$u_{n+1} = \frac{1}{2} \left( u_n + \frac{2}{u_n} \right).$$

On admet que la suite  $(u_n)$  converge vers  $\sqrt{2}$  (on verra comment démontrer cela cette année). Le but de cet exercice est d'implémenter notre propre méthode d'approximation de  $\sqrt{2}$ .

**Remarque 2.** On peut stocker une excellente approximation du nombre  $\sqrt{2}$  dans la variable  $r$  à l'aide du script suivant :

```
1 from math import sqrt
2 r = sqrt(2)
```

**Q1** Écrire une fonction `suite` qui prend en argument un entier  $n$  et qui renvoie  $u_n$  le  $n$ -ème terme de la suite.

**Q2** Écrire une fonction `seuil` qui prend en argument un nombre réel  $\varepsilon$  (représentant un écart) et qui renvoie la plus petite valeur de  $n$  telle que  $|u_n - \sqrt{2}| \leq \varepsilon$ .

**Q3** Calculer une approximation de  $\sqrt{2}$  à  $10^{-2}$  près en utilisant la fonction `seuil`. Pour quel terme de la suite  $(u_n)$  cette approximation est-elle atteinte?

**Q4** Calculer une approximation de  $\sqrt{2}$  à  $10^{-4}$  près en utilisant la fonction `seuil`. Pour quel terme de la suite  $(u_n)$  cette approximation est-elle atteinte?

## Exercice 5 La suite de Syracuse

Soit  $f : \mathbb{N} \rightarrow \mathbb{N}$  la fonction définie par

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

On appelle suite de Syracuse de l'entier  $N$  la suite  $(u_n)$  définie par  $u_0 = N$  et  $\forall n \in \mathbb{N}, u_{n+1} = f(u_n)$ .

**Q1** Calculer à la main les termes de la suite de Syracuse de l'entier 3. Que se passe-t-il lorsque la suite atteint la valeur 1 ?

**Remarque 3.** 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  $k$  et renvoyant  $f(k)$ .

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

**Q4** Écrire une fonction `temps_de_vol` prenant en argument un entier  $N$  et renvoyant le premier entier  $n$  tel que  $u_n = 1$ , où la suite  $(u_n)$  est la suite de Syracuse de l'entier  $N$ . On vérifiera que `temps_de_vol(15)=17`.

**Q5** Tracer les valeurs de `temps_de_vol(N)` pour  $1 \leq N \leq 1000$ .

**Remarque 4.** Pour tracer des graphiques :

— Taper l'instruction

```
1 import matplotlib.pyplot as plt
```

au tout début du fichier.

— On pourra dessiner les points du graphique un par un en utilisant la commande `plt.plot(x,y,'bo')` pour tracer un point bleu aux coordonnées  $(x, y)$ .

— Utiliser la commande

```
1 plt.show()
```

à la toute fin du script.

## Exercice 6 Des sauts de puce

Imaginons une puce se déplaçant aléatoirement sur une ligne allant soit en avant soit en arrière (pas de 1 ou -1). Par exemple, si cette dernière est à l'emplacement 2, elle peut sauter aléatoirement à l'emplacement 1 ou 3.

**Q1** Avec une boucle `while`, écrire une fonction simulant le mouvement de cette puce de l'emplacement initial 0 à l'emplacement final 5. La fonction prendra en argument l'emplacement d'arrivée et renverra le nombre de sauts effectués par la puce.

**Remarque 5.** Pour simuler le comportement aléatoire, on utilisera

```
1 import random # A positionner en debut de script
2 x = random.choice([-1,1]) # A positionner dans votre script
```

Après ce script, la variable `x` a 1 chance sur 2 d'être égale à -1, et 1 chance sur 2 d'être égale à 1.

- Q2** Combien de sauts sont nécessaires pour réaliser ce parcours? Obtient-on le même résultat à chaque fois?
- Q3** Aurait-on pu réaliser cette simulation avec une boucle `for`?
- Q4** Écrire un programme affichant les résultats de plusieurs essais de la simulation.
- Q5** Calculer le nombre moyen de sauts effectués sur ces essais.