

TP7

CORRIGÉ TP6 – LISTES SIMPLES

À préparer : Cours sur les listes + exercice 2

Notions utilisées dans ce TP

Ce TP utilise les parties suivantes du vade-mecum (en plus des TP précédents).

- II : listes
- III : bibliothèques math et random

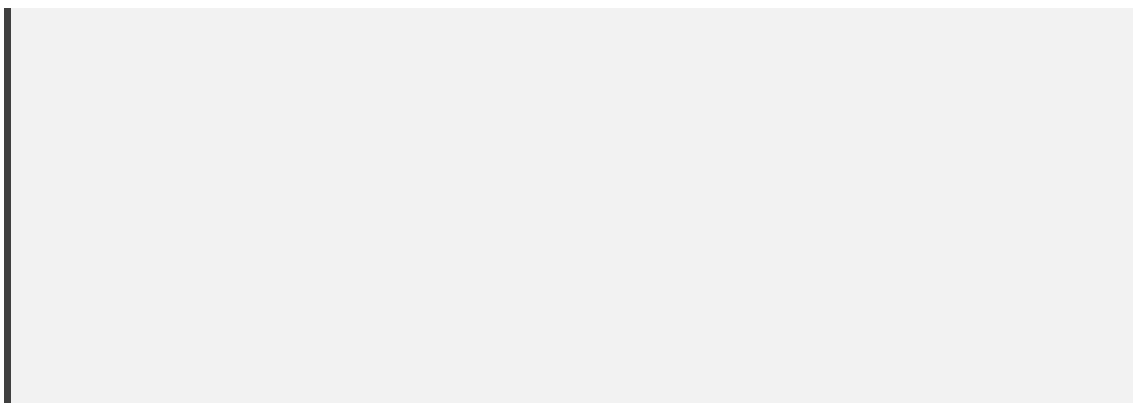
Exercice 1 On appelle suite de Syracuse toute suite d'entiers naturels $(u_n)_{n \in \mathbb{N}}$ définie de la manière suivante :

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

1. Compléter le tableau suivant avec les valeurs de la suite de $u_0 = 10$ à u_3 .

$u_0 = 10$
$u_1 =$
$u_2 =$
$u_3 =$

2. Écrire une **fonction** python **Syracuse(u0,n)** qui, étant donné un nombre $u_0 \in \mathbb{N}^*$ et un entier naturel n , renvoie la valeur de u_n .



3. Testez votre fonction en vérifiant la cohérence avec la question 1.
On calculera aussi u_{15} pour $u_0 = 145$.

4. La conjecture de Syracuse est une hypothèse selon laquelle toute suite de Syracuse atteint 1. On supposera que celle-ci est vraie (cette conjecture n'a jamais été démontrée jusqu'à présent).

On appelle **Temps de Vol** le plus petit indice n tel que $u_n = 1$.

Il est de 17 pour la suite de Syracuse avec $u_0 = 15$ et de 46 pour la suite de Syracuse avec $u_0 = 127$.

A l'aide d'un programme utilisant un while, vérifier l'affirmation précédente.

```
u0 = int(input('Saisir la valeur de u0 :'))
```

Exercice 2 Manipulation simple de listes.

Tout cet exercice peut être fait dans la console.

1. Définir la liste L des nombres 5, 6, -1, 2 et 4 (sans cet ordre).

2. Afficher le premier élément de L .

3. Afficher le troisième élément de L (le -1 donc).

4. Modifier le deuxième élément de L (le 6) en un 10.

5. Afficher le nombre d'éléments de L .

6. Ajouter un 7 à la fin de L .

7. Ajouter un -5 au début de L .

8. Supprimer le dernier élément de L .

9. Vérifier que 7 n'est plus dans la liste L (avec un test booléen).

10. Créer une liste $T = [2, 3, 4, \dots, 15]$ avec un range.

11. Afficher T . Pour avoir un visuel explicite, il faut convertir T en liste (un range n'est pas tout à fait une liste pour python).

```
>>> list(T)
```

12. Créer une liste $U = [6, 8, 10, \dots, 30]$ avec un range puis l'afficher pour vérifier.

13. Vérifier, avec un booléen, que 20 est dans U .

Exercice 3 Écrire une fonction Python qui, étant donnés trois flottants a, b et c , crée et renvoie la liste contenant ces trois nombres et ordonnée de façon croissante. *Interdiction d'utiliser une fonction de tri toute faite.*

Exercice 4 Écrire une fonction `second_degre(a,b,c)` qui étant donnés trois nombres a, b, c avec $a \neq 0$, renvoie la **liste des solutions réelles** de l'équation $ax^2 + bx + c = 0$.

```
def second_degre(a,b,c):  
    '''second_degre(a, b, c) -> liste des solutions de ax**2+bx+c=0  
    a,b,c sont des entiers ou flottants, a est non nul.'''  
  
    assert ..... # vérification, a non nul.
```