

On utilisera un seul fichier Python nommé TD3.py pour l'ensemble du TD

Formulaire :

while a != 3 :	tant que a est différent de 3
15/2	renvoie 7.5 (division à virgules)
14/2	renvoie 7.0 (division à virgules)
15//2	renvoie 7 (quotient dans une division euclidienne)
15%2	renvoie 1 (reste dans une division euclidienne)

On considère la suite de Syracuse définie par :

$$\begin{cases} U_0 = a > 0 & \text{entier fixé} \\ U_{n+1} = \frac{U_n}{2} & \text{si } U_n \text{ est pair} \\ U_{n+1} = 3U_n + 1 & \text{si } U_n \text{ est impair} \end{cases}$$

La conjecture de Syracuse annonce qu'à partir de n'importe quelle valeur de départ, la suite finit toujours par retomber sur le cycle (4 ; 2 ; 1).

On appelle **temps de vol** le nombre de valeurs prises avant de retomber à 1 et **l'altitude** la valeur maximale de la suite.

1°) Calculer, à la main, les quinze premiers termes de la suite lorsque $U_0 = 10$. Préciser le temps de vol et l'altitude correspondants.

2°) Ecrire un algorithme nommé syracuse(u,n) qui, aux entiers u et n fixés, renvoie la liste des n premiers termes de la suite de Syracuse avec $U_0 = u$

Tester avec $u = 10$ et $n = 15$ puis avec différentes valeurs de u et de n (n suffisamment grand pour obtenir le cycle (4 ; 2 ; 1))

3°) Ecrire un algorithme nommé tempsvol(u) qui, à un entier u fixé, renvoie le temps de vol de la suite de Syracuse avec $U_0 = u$

Tester avec $u = 10$ puis avec d'autres valeurs de u

4°) En utilisant **les deux algorithmes précédents**, écrire un programme nommé altitude(u) qui, à un entier u fixé, renvoie l'altitude de la suite de Syracuse avec $U_0 = u$

Tester avec $u=10$ puis avec d'autres valeurs de u