

Suites de Syracuse

En langage Python, l'écriture **a%b** permet de renvoyer le reste de la division euclidienne de **a** par **b** (où **a** et **b** sont des nombres entiers positifs, **b** non nul).

0) Question préliminaire :

Si **a** est une variable contenant un nombre entier positif :

- Quelles sont les valeurs que peut renvoyer la saisie ci-contre ? `>>> a%2`
- A quelles propriétés du nombre **a** correspondent chacune de ces valeurs ?

Définition de la suite de Syracuse associée à un nombre **a** :

A partir d'un entier non nul **a**, on peut construire une suite de nombres de la façon suivante :

$$u_0 = a \quad \text{et} \quad u_{n+1} = \begin{cases} \frac{u_n}{2} & \text{si } u_n \text{ est pair} \\ 3u_n + 1 & \text{si } u_n \text{ est impair} \end{cases}$$

(chaque terme de la suite est obtenu en divisant le précédent par 2 si celui-ci est pair, et en le multipliant par 3 et en ajoutant 1 s'il est impair)

1) Calculer, à la main, les 6 premiers termes de la suite de Syracuse associée au nombre 17.

2) Ecrire une fonction Python **suiv_Syracuse(p)** :

- qui reçoit en argument un terme **p** d'une suite de Syracuse ;
- qui renvoie le terme suivant de la suite.

NOTE : Pour s'assurer que la valeur renvoyée soit de type `int`, on pourra écrire la division sous la forme `//` qui renvoie le quotient entier d'une division.

3) On considère la fonction Python **Deb_Syracuse(a)** suivante :

```
def Deb_Syracuse(a) :  
    L=[a]  
    for k in range(5) :  
        a= suiv_Syracuse(a)  
        L.append(a)  
    return L
```

a) Compléter le tableau suivant avec les valeurs prises successivement par les variables, si on appelle la fonction **Deb_Syracuse(a)** avec **a=7**.

k	a	L
	7	[7]
0		
1		
2		
3		
4		

b) Coder cette fonction et vérifier que la liste renvoyée par l'instruction `>>> Deb_Syracuse(7)` est cohérente avec votre tableau.

c) Quelle est l'utilité de cette fonction **Deb_Syracuse(a)** ?

4) Ecrire une fonction Python **Tab_Syracuse(a,N)** :

- qui reçoit en arguments le 1^{er} terme **a** d'une suite de Syracuse et un nombre entier **N** ≥ 1 ;
- qui renvoie la liste des **N** premiers termes de cette suite de Syracuse.

Notion de vol associé à un nombre a :

On appelle **vol** correspondant à a , la liste des valeurs obtenues par la suite de Syracuse à partir de a , et s'arrêtant au premier terme valant 1. (*)

On appelle **durée du vol** le nombre de termes de la liste, et on appelle **altitude maximale** la plus grande valeur de cette liste.

- 5) Compléter, à la main, la suite de nombres obtenus à la question 1) pour obtenir le vol correspondant au nombre 17.
Quelle est la longueur de ce vol ? Quelle est l'altitude maximale de ce vol ?

- 6) Ecrire une fonction Python **Vol_Syracuse(a)** :
- qui reçoit en argument le 1^{er} terme a d'une suite de Syracuse
 - qui renvoie la liste correspondant au vol obtenu avec a .

- 7) Saisir la série d'instructions suivantes, et expliquer, pour chacune d'elle, ce que représente le résultat obtenu.

```
>>> v=Vol_Syracuse(137)
```

```
>>> v
```

```
>>> max(v)
```

```
>>> len(v)
```

8) Prolongements possibles :

- a) Ecrire une fonction Python qui renvoie la plus petite valeur a dont le vol atteint une altitude au moins égale à 150.

Adapter la fonction pour qu'elle renvoie la plus petite valeur a dont le vol atteint une altitude au moins égale à M , où M est passé en argument.

- b) Ecrire une fonction Python qui renvoie la plus petite valeur a dont la durée de vol est supérieure à 40.

Adapter la fonction pour qu'elle renvoie la plus petite valeur a dont la durée de vol est supérieure à T , où T est passé en argument.

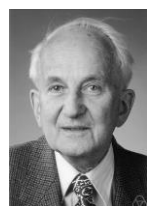
- c) Ecrire une fonction Python qui renvoie la valeur de a inférieure à 100000 pour laquelle le vol est le plus long.

Adapter la fonction pour qu'elle renvoie la valeur de a inférieure à N pour laquelle la durée de vol est maximale, où N est passé en argument.

- d) Ecrire une fonction Python qui renvoie la valeur de a inférieure à 100000 pour laquelle l'altitude atteinte est maximale.

Adapter la fonction pour qu'elle renvoie la valeur de a inférieure à N pour laquelle l'altitude atteinte est maximale, où N est passé en argument.

(*) La conjecture de Syracuse stipule que quelle que soit la valeur a choisie, la suite de Syracuse finira par « atterrir », c'est-à-dire qu'elle atteindra au bout d'un nombre fini d'itérations la valeur 1. A ce jour, cette conjecture n'a jamais été démontrée, mais elle a été vérifiée pour tous les entiers inférieurs à $2^{62} \approx 4,6 \times 10^{18}$... avec des ordinateurs évidemment.



Lothar Collatz (1910 – 1990)
est à l'origine de cette conjecture