**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).

1. *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 :
 $u\_{0}=a$ et $u\_{n+1}=\left\{\begin{matrix}\frac{u\_{n}}{2} si u\_{n} est pair\\3u\_{n}+1 si u\_{n} est impair\end{matrix}\right.$

*(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.

*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.*

1. 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.
1. 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**

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



1. Coder cette fonction et vérifier que la liste renvoyée par l’instruction **>>> Deb\_Syracuse(7)** est cohérente avec votre tableau.
2. Quelle est l’utilité de cette fonction **Deb\_Syracuse(a)** ?
3. Ecrire une fonction Python **Tab\_Syracuse(a,N)** :
* qui reçoit en arguments le 1er terme **a** d’une suite de Syracuse et un nombre entier **N**$\geq $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.

1. 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 ?

1. Ecrire une fonction Python **Vol\_Syracuse(a)** :
* qui reçoit en argument le 1er terme **a** d’une suite de Syracuse
* qui renvoie la liste correspondant au vol obtenu avec **a**.
1. 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)**

1. **Prolongements possibles :**
2. 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.

1. 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.

1. 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.

1. 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 $2^{62}≈4,6×10^{18}$… avec des ordinateurs évidemment.

**Lothar Collatz** (1910 – 1990)

est à l’origine de cette conjecture