

TP de Langages de script n° 2 bis : Exercices additionnels.

Exercice 1 : suite de Syracuse

La suite de Syracuse d'origine N est définie par $u_0 = N$ et, pour tout $n \geq 0$,

$$u_{n+1} = \begin{cases} \frac{1}{2}u_n & \text{si } u_n \text{ est pair} \\ 3u_n + 1 & \text{sinon.} \end{cases}$$

Une conjecture célèbre affirme que, pour toute valeur initiale N , la suite a le même comportement à la limite : elle boucle sur les valeurs 4, 2 et 1. On appelle *temps de vol* de la suite la plus petite valeur n telle que $u_n = 1$, et *altitude* de la suite la plus grande valeur qu'elle atteint.

Écrire une fonction calculant ces deux valeurs en fonction de la valeur initiale de la suite.

Exercice 2 : encore des suites

On considère quatre entiers a , b , x_0 et x_1 et la suite $(x_n)_n$ définie, pour tout $n \geq 0$, par $x_{n+2} = ax_{n+1} + bx_n$.

1. À l'aide de l'interpréteur PYTHON, afficher toutes les valeurs de $(x_n)_n$ pour $a = 2$, $b = 3$, $x_0 = 1$ et $x_1 = 3$ pour n allant de 1 à 20 en utilisant une boucle **while**.
2. Écrire une fonction itérative utilisant une boucle **while** qui prend en entrée 5 arguments correspondant à a , b , x_0 , x_1 et au rang n du terme à calculer, et renvoie la valeur de x_n .
3. Même question avec une fonction itérative utilisant une boucle **for**.
4. Même question avec une fonction récursive. Testez votre fonction pour $a = 1$, $b = 1$, $x_0 = 1$, $x_1 = 1$ et $n = 40$. Si le résultat ne s'affiche pas en un temps raisonnable, comment réparer votre fonction ?

Exercice 3 : système de fichiers

1. À l'aide des modules **os** et **os.path**, créer un script **taille.py** qui calcule la taille totale de tous les fichiers contenus dans le répertoire courant.
2. Modifier ce script pour la taille totale d'une arborescence, *sous-répertoires compris*. Vous pouvez utiliser la fonction **walk** du module **os**. Pour avoir une idée du comportement attendu, tester la commande « **du -b -s** » dans un terminal.
3. Modifier enfin ce script pour afficher individuellement la taille de chaque sous-répertoire, et plus seulement le total. Pour avoir une idée du comportement attendu, tester la commande « **du -b** » dans un terminal.

Exercice 4 : un peu de rangement

Avertissement : Attention à ne pas effacer fortuitement vos fichiers !

On souhaite isoler les fichiers PYTHON (avec extension PY) présents dans une arborescence et homogénéiser leurs noms.

1. Écrire un script listant tous les fichiers PYTHON contenus dans un répertoire.
2. Modifier le script pour qu'il recopie tous ces fichiers dans un nouveau répertoire.
3. Modifier le script pour que, dans ce nouveau répertoire, tous les noms de fichiers soient en minuscules.
4. Modifier le script pour qu'il traite toute l'arborescence ayant une racine donnée, et stocke les copies des fichiers dans une copie de l'arborescence.