

TP 5 Python – Exercices bilan

Python

Exercice 1. On définit une suite $(u_n)_{n \in \mathbb{N}}$ par $u_0 = 42$ et, pour tout $n \in \mathbb{N}$, $u_{n+1} = \frac{4u_n - 9}{u_n - 2}$.

1. Écrire une fonction Python qui calcule le n -ième terme de la suite u_n .
2. Vérifiez avec votre fonction que $u_6 \approx 3.1659574$
3. À l'aide de votre fonction, conjecturez la limite de la suite.
4. Écrire un script qui détermine à partir de quel rang n la valeur u_n est proche de la limite de la suite à moins de 0,0001.

Exercice 2. Écrire une fonction qui prend en paramètre trois nombres u_0 , q et n (avec $n \in \mathbb{N}$), qui renvoie le terme d'indice n de la suite géométrique de premier terme u_0 et de raison q , ainsi que la somme de tous les termes entre u_0 et u_n .

Exercice 3. Écrire un programme qui calcule et affiche la somme de tous les multiples de 29 s'écrivant avec au plus 3 chiffres.

On écrit deux versions de ce programme, l'une avec une boucle `for` et l'autre avec une boucle `while`.

Exercice 4. Écrire une fonction qui prend en paramètre une liste, et renvoie un booléen qui indique si la liste est triée par ordre croissant (large) ou non.

Exercice 5. Écrire une fonction qui prend en paramètre une liste, et qui la "retourne", c'est-à-dire qui renvoie une liste contenant les mêmes valeurs, mais dans l'ordre inverse.

Exercice 6. On plie plusieurs fois une feuille de papier de format A4 (21 cm times 29.7 cm) et d'épaisseur 0.01 cm , et on veut calculer les dimensions de cette feuille après un certain nombre de pliages.

1. Écrire une fonction de paramètre n , qui renvoie les dimensions (longueur, largeur et épaisseur) de la feuille après n pliages. Tester pour 5 pliages, puis pour 10 pliages.
2. Écrire une fonction, qui prend en paramètre une hauteur h (en cm), et qui renvoie le nombre de pliages nécessaires pour que l'épaisseur finale du papier soit supérieure à h .
Tester pour une hauteur de $2,5\text{ m}$, puis pour la distance Terre-Lune (environ $380\,400\text{ km}$).

Exercice 7. 1. Écrire une fonction `est_diviseur`, qui prend en argument deux entiers a et b , et qui renvoie un booléen qui indique si a est ou non un diviseur de b .

2. Écrire une fonction `est_premier`, qui prend en argument un entier n , et qui renvoie un booléen qui indique si n est premier ou non.

3. Chercher sur Internet le fonctionnement du Crible d'Ératosthène.

Écrire une fonction `crible_eratosthene(n)`, qui prend en argument un entier naturel n , et qui renvoie une liste L de taille n telle que pour tout k entre 0 et $n - 1$, $L[k]$ est un booléen :

- `True` si k est premier ;
- `False` sinon.