

DS1 Informatique

PC Lycée Lavoisier

17 Novembre 2023

Consignes :

1. Écrire proprement.
2. Bien indenter vos codes (deux gros carreaux ou quatre petits carreaux).
3. Vous avez le droit d'utiliser une fonction écrite à une question précédente même si vous n'avez pas réussi cette question.

Les classiques (ne soyez donc pas trop baroques dans vos réponses !)

1. Écrire une fonction `DicoOccurrences(L)` qui prend comme paramètre une liste `L` et qui renvoie le dictionnaire des occurrences c'est-à-dire que les clés sont les éléments de la liste et la valeur d'une clé est le nombre d'occurrences de la clé dans la liste `L`.
2. Écrire une fonction `Moyenne(L)` où `L` est une liste de listes de flottants ou d'entiers et qui renvoie la moyenne. Ainsi, pour `L=[[6,1,7],[9],[2,5]]`, la fonction renvoie 5.
3. Soit n et p deux entiers naturels non nuls. Donner une commande Python qui crée une liste à n éléments tel que chaque élément soit une liste de p éléments remplies de 0.

Nombres amicaux (comme ce sujet donc !)

1. Soit $(n, d) \in (\mathbb{N}^*)^2$, quelle commande Python permet de savoir si d divise n .
2. Écrire une fonction Python `ListeDiviseurs(n)` qui étant donné un entier $n \in \mathbb{N}^*$ renvoie la liste des diviseurs (positifs) de n . Proposer deux méthodes pour créer cette liste. Par exemple, pour $n = 6$, la fonction renverra la liste `[1,2,3,6]`.
3. Écrire une fonction `SommeDiviseurs(n)` qui renvoie la somme des diviseurs positifs stricts de $n \in \mathbb{N}^*$ (tous les diviseurs de n sauf n lui-même). Par exemple, pour $n = 6$, la fonction renverra 6 (car $1 + 2 + 3 = 6$).

Un entier $n \geq 1$ est dit **parfait** s'il est égale à la somme de ces diviseurs stricts. Par exemple 6 est parfait car $1 + 2 + 3 = 6$, 8 n'est pas parfait car $1 + 2 + 4 \neq 8$.

4. Écrire une fonction `parfait(n)` qui renvoie `True` si n est parfait et `False` sinon.

On dit que deux entiers naturels non nuls a et b sont **amicaux** si a est la somme des diviseurs stricts de b et si b est la somme des diviseurs stricts de a avec $a \neq b$.

5. Écrire un programme qui renvoie a et b deux nombres amicaux tel que $\max(a, b)$ soit le plus petit possible.
Veiller à ne pas utiliser plus d'une fois la fonction `SommeDiviseurs(k)` pour le même k et pour ce faire, on pourra utiliser un dictionnaire.

Anagrammes (ne vous emmêlez donc pas dans les lettres ni dans le code !)

On rappelle qu'un mot un anagramme d'un autre mot si on peut passer d'un mot à un autre en permutant les lettres : `chien` et `niche` sont donc des anagrammes. Étant donné une chaîne de caractères nommé `s`, on peut créer une fonction `DicoOccurrences(s)` qui renvoie le dictionnaire des occurrences : les clés sont les lettres dans `s` et la valeur d'une clé est le nombre d'occurrences. Le code étant très similaire à la question 1 de la première partie, on ne demande pas de recoder cette fonction. On dispose aussi d'une liste nommée `Liste` contenant tous les mots français.

1. Que renvoie la fonction `DicoOccurrences(s)` si `s` a pour valeur `"anagrammes"` ?
2. Étant donnés deux mots, comment peut-on utiliser la fonction `DicoOccurrences` pour savoir si les deux mots sont anagrammes l'un de l'autre ?
3. Créer une fonction `DicoDico(Liste)` qui renvoie un dictionnaire dont les clés seront les mots de `Liste` et la valeur associée à une clé `s` sera le résultat de `DicoOccurrences(s)`.
4. En déduire un script permettant de trouver le ou les mots ayant le plus d'anagrammes.