

DS - Moyennes

Consignes

Les consignes suivantes **doivent** être respectées sous peine d'être pénalisé par des **points négatifs**.

Le code doit être clair.

- Évitez les lignes à rallonge. S'il faut introduire une variable intermédiaire, faite le.
- Nommez vos variables avec du sens. Évitez toutefois des noms trop longs. Auquel cas décrivez le rôle de la variable en commentaire.
- Ne surchargez pas de commentaires. Les commentaires utiles sont :
 - Spécification de la fonction
 - Description du rôle d'une variable
 - Description du rôle d'une boucle complexe.

Soignez la présentation de votre code.

- Marquez clairement les indentations.
- Éviter les ratures trop nombreuses dans un code. Une ou deux maximum.
- Écrivez votre instruction sur plusieurs lignes si elle est longue en ajoutant simplement une indentation après le retour à la ligne.

Respectez la syntaxe Python. - 0 à l'exercice si syntaxe inventée.

- Vérifiez que ce que vous écrivez respecte la syntaxe.
- Les petites erreurs n'impliquent pas 0 (oublie de :, = au lieu de ==, ...)
- Si vous avez une idée, mais que vous n'arrivez pas à l'écrire, n'inventez pas. Écrivez votre idée en commentaire.

Pour nommer une fonction utilisez le nom donné dans l'énoncé - 0 à l'exercice sinon

Utilisation de print ou de input interdite pour le moment - 0 à l'exercice sinon

Vous avez le droit d'utiliser une fonction implémentée dans une question précédente, même si vous n'avez pas réalisé la question.

Rappels de Python : Si L est une liste, alors $L[i]$ désigne le i -ème élément de cette liste et $\text{len}(L)$ la longueur de la liste. La commande $L[i] = x$ affecte la valeur de l'expression x au i -ème élément de la liste L . L'expression $[]$ construit une liste vide. L'expression $n * [x]$ construit une liste de longueur n contenant n occurrences de x . La commande $L.append(x)$ modifie la liste associée à L en lui rajoutant un nouvel élément final contenant x . La commande $L.pop()$ modifie la liste L en supprimant son dernier élément et en renvoyant sa valeur. Si $L1$ est une autre liste, $L + L1$ renvoie la concaténation de L et $L1$. $x \text{ in } L$ renvoie un booléen qui vaut vrai si et seulement si x est un élément de L .

Si T est un n -uplet, alors $T[i]$ désigne le i -ème élément de ce n -uplet. $(1, 2)$ permet de créer le couple $(1, 2)$.

Important : Seules les opérations sur les listes apparaissant dans le paragraphe précédent sont autorisées dans les réponses. Si une fonction Python standard est nécessaire, elle devra être réécrite.

1 Exercices

1.1 Cours

Question 1.1.

Donnez l'expression qui permet d'accéder à la dernière lettre de la chaîne de caractère `s`.

Question 1.2.

Que renvoient l'expressions `(1, 2, 3) + (4, 5, 6)` ?

Question 1.3.

Donnez l'instruction permettant de renvoyer la liste des ligne du fichier associé à l'objet fichier `nadine`.

Question 1.4.

Qu'est que fait le code ci-dessous ?

```
s = (1, 2, 3)
s[1] = 4
```

1.2 Algorithmes Classiques

Question 1.5.

La suite de Fibonacci est définie par récurrence double :

$$F_0 = 0, F_1 = 1, F_{n+2} = F_{n+1} + F_n$$

Ecrire une fonction `syracuse` avec la spécification suivante :

Entrées : un entier $n \geq 0$

Sortie : F_n

Question 1.6.

Écrire une fonction `recherche_lineaire` avec la spécification suivante :

Entrées : une liste `l` et un élément `x`

Sortie : L'indice de la première occurrence de `x` dans `l`, ou -1 si `x` n'est pas dans `l`

Votre fonction devra utiliser une boucle `for` pour parcourir les éléments de la liste

Question 1.7.

Écrire une fonction `moyenne` avec la spécification suivante :

Entrées : une liste d'entiers `L`

Sortie : la moyenne des valeurs de la liste `l`

2 Problèmes

2.1 Moyennes

Pour les trois premières question on considèrera que les notes d'un élèves sont données par une liste de couples (n, c) avec n un flottant représentant une note entre 0 et 20 et c un coefficient entier.

Question 2.1.

Implémentez une fonction `plus_faible` qui prend en entrée deux couples de notes et coefficient, (n_1, c_1) et (n_2, c_2) dans cet ordre, et qui renvoie un booléen, qui est vrai si et seulement si $n_1 c_1 < n_2 c_2$

Question 2.2.

Implémentez une fonction `meilleure` qui prend en entrée une liste de notes avec coefficient et qui renvoie la moins faible dans le sens de la fonction `plus_faible` implémentée à la question précédente.

Question 2.3.

Implémentez une fonction `moyenne pondérée` qui prend en entrée une liste de notes avec coefficient et qui renvoie la moyenne pondérée de la liste. La moyenne pondérée est la somme des notes multipliée par leur coefficient, le tout divisé par la somme des coefficients.

Question 2.4.

Pour cette question on considèrera que les notes d'un élèves sont données par *deux* listes. La première liste N de flottants représente une liste de note entre 0 et 20. La second liste, C fait la même taille que N et contient à la position $0 \leq i \leq \text{len}(L) - 1$ le coefficient de la note $N[i]$.

Implémentez une fonction `moyenne pondérée` qui prend en entrée deux listes représentant les notes avec coefficient comme décrit ci-dessus, et qui renvoie la moyenne pondérée de la liste.

2.2 Distance

Les points du plan seront représentés par des couples de flottant (x, y) tel que x est l'abscisse du point et y dont ordonnée. On rappelle qu'étant donné un couple p de flottant, les instructions $x = p[0]$ et $y = p[1]$ permette respectivement de récupérer l'abscisse et l'ordonnée du point représenté par p .

Étant donnés deux points A et B on rappelle la définition de la distance euclidienne entre A et B , notée AB :

$$AB = ((x_A - x_B)^2 + (y_A - y_B)^2)^{\frac{1}{2}}$$

Question 2.5. *Distance*

Implémenter une fonction `distance` qui prend en entrée deux points sous la forme de deux couples de flottant et qui renvoie la distance euclidienne entre ces deux points.

Question 2.6. *Barycentre*

On considère $n + 1$ points du plan, $A_0(x_0, y_0), \dots, A_n(x_n, y_n)$. Le barycentre de ces points est le point :

$$B = \left(\frac{\sum_{i=0}^n x_i}{n+1}, \frac{\sum_{i=0}^n y_i}{n+1} \right)$$

Implémenter une fonction `barycentre` qui prend en entrée une liste P de points et qui renvoie un le barycentre de la liste.

Question 2.7. *Plus proches*

Implémentez une fonction `plus_proche` qui prend en entrée une liste de points l de taille au moins 2 et renvoie les deux points de l dont la distance est la plus faible.

Si il y a plus qu'une paire de points dont la distance est égale à la plus faible, renvoyez la paire contenant le point le plus éloigné dans la liste (d'indice le plus grand).