

CC4 - sujet section B

Info1.Algo1 - 2022-2023 Semestre Impair

Durée de l'épreuve : 1h

- Pour chaque exercice vous devez déposer un fichier correspondant à votre travail.
- Vous ne devez pas modifier le nom du fichier.
- Un fichier qui ne compile pas, ou qui ne passe pas les fonctions de tests proposées dans le sujet ne sera pas corrigé et entraînera la note de 0 à l'exercice, sans aucune investigation supplémentaire.
- Il est naturellement interdit de modifier la fonction de tests qui vous est proposée, vous devez être particulièrement attentif à nommer correctement vos fonctions et à respecter les paramètres attendus.
- Le passage des différents tests proposés avec le sujet n'est en aucune façon une garantie de la bonne correction de votre travail, des tests complémentaires seront passés a posteriori.

Exercice 1

On appelle matrice transposée une matrice dont on permute l'organisation ligne/colonne.

Exemple :

Si la matrice est :

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{pmatrix}$$

... alors sa matrice transposée est :

$$\begin{pmatrix} 1 & 5 & 9 \\ 2 & 6 & 10 \\ 3 & 7 & 11 \\ 4 & 8 & 12 \end{pmatrix}$$

Dans le fichier **ex01.py**, compléter la fonction **transposer** qui accepte en paramètre une matrice et retourne sa matrice transposée.

Exercice 2

Dans le fichier **ex02.py**, compléter le corps de la fonction *récursive* `somme_chiffre_impairs` qui accepte en paramètre un entier `n` et retourne la somme de ses chiffres impairs.

Exemples :

- La somme des chiffres impairs de 123456 est $1+3+5=9$.
- La somme des chiffres impairs de 42 est 0.

Exercice 3

Dans le fichier **ex03.py**, compléter la fonction *récursive* `diviser_par_deux` qui accepte un paramètre `arbre` et qui retourne un arbre de même structure où les valeurs impaires n'ont pas été modifiées et les valeurs paires ont été divisées par 2.

On supposera, sans le vérifier dans la fonction, que les valeurs de l'arbre sont des entiers positifs.

On rappelle que la division **entière** de deux entiers `a` et `b` est obtenue par `a // b`.

On considérera que `diviser_par_deux(creer_arbre_vide())` retourne un arbre vide.

Exemple : Si l'arbre est :

```
┌ 4 ┐  
5   4
```

`diviser_par_deux(arbre)` retourne :

```
┌ 2 ┐  
5   2
```

Exercice 4

Dans le fichier **ex04.py**, écrire la fonction *récursive* `partitionner_negatifs_positifs` qui accepte en paramètre une variable `liste` composée d'entiers et retourne deux listes chaînées `liste_negatifs` et `liste_positifs` composées respectivement des éléments de `liste` qui sont strictement inférieurs à 0 et de ceux qui sont supérieurs ou égaux à 0.

L'ordre des éléments dans les listes chaînées résultantes doit être le même que dans la liste chaînée `liste`.

On supposera, sans le vérifier dans la fonction, que les éléments de la liste sont des entiers.

Exemple : Si la liste chaînée `liste` est `(-8,(10,(17,(-9,None))))` alors `partitionner_negatifs_positifs(liste)` retourne le tuple composé de `(-8,(-9,None))` et `(10,(17,None))`.