TD 1 bis : retour sur la manipulation de listes

Les sujets de TD ne sont ni relevés, ni notés. Ils sont volontairement conçus pour ne pas pouvoir être traités en une seule séance ; n'hésitez cependant pas, si vous le souhaitez, à tenter de les finir chez vous.

En complément du TD1, l'objectif de ce TP « 1 bis » est de revenir rapidement sur quelques algorithmes de manipulation des listes.

Les questions suivantes sont indépendantes.

1. Écrire une fonction prenant en entrée une liste et renvoyant la même liste « retournée » (c'est-à-dire dans l'autre sens).
2. Écrire une fonction prenant comme unique argument une liste contenant des nombres réels, renvoyant `True` si ces nombres sont triés dans l'ordre croissant, `False` sinon. L'on tentera de limiter le nombre d'itérations superflues.
3. (a) Écrire cinq fonctions prenant comme argument une liste de nombres réels et renvoyant respectivement la somme des éléments, le plus petit élément, le plus grand élément, l'indice du plus petit élément et l'indice du plus grand élément de ces listes.
   (b) Modifier ces fonctions de sorte qu'elles prennent un deuxième paramètre, devant être une liste de booléens de même taille que la première. Les fonctions doivent alors renvoyer respectivement somme, plus petit élément, plus grand élément, indice du plus petit élément et indice du plus grand élément *sur les éléments de la première liste dont l'indice correspond à `True` dans la seconde uniquement.* Par exemple, pour les paramètres `[1, 42, 0, -7, 3, 5, -31]` et `[True, False, False, True, True, True]`, les fonctions doivent renvoyer respectivement -22, 5, -31, 5 et 6. Une exception de type `ValueError` doit être soulevée si les deux listes ont une longueur incompatible.
   (c) Rendre le deuxième argument facultatif, et faire en sorte qu'il soit par défaut une liste de `True` ayant la même longueur que le premier paramètre.
4. Écrire une fonction prenant en entrée deux listes à deux dimensions représentant des matrices d'entiers de taille $n \times p$ et $p \times m$, et renvoyant leur produit sous la forme d'une liste bidimensionnelle de taille $n \times m$.
5. *(Pour les plus courageux.)* L'on représente par la suite les matrices carrées sous la forme de listes bidimensionnelles.
   (a) Écrire une fonction prenant comme paramètre une matrice carrée d'ordre n et deux indices de lignes `i` et `j`, et modifiant la matrice en effectuant une permutation $L_i \leftrightarrow L_j$. Cette fonction ne doit rien renvoyer.
   (b) De même, écrire une fonction prenant comme paramètre une matrice, deux indices de lignes `i` et `j` et un nombre réel `alpha`, et effectuant la transvection $L_j \leftarrow L_j + \alpha L_i$.
   (c) Enfin, écrire une fonction prenant comme paramètre une matrice, un indice de ligne `i` et un nombre réel non nul *alpha* et effectuant la dilatation $L_i \leftarrow \alpha L_i$.
   (d) Écrire une fonction prenant comme paramètre une matrice carrée d'ordre n représentée sous la forme d'une liste bidimensionnelle, et renvoyant son inverse calculé à l'aide de la méthode du pivot de Gauss. L'on soulèvera une exception de type `ValueError` si la matrice d'entrée n'est pas inversible. La matrice passée en argument ne doit pas être modifiée, il est donc nécessaire d'en faire une copie.
6. Écrire une fonction résolvant le système matriciel $AX = B$ d'inconnue X si celui-ci admet une solution unique.