

TP4 : Fonctions sur les listes et les tableaux

Ce TP permettra de mettre en pratique la manipulation de listes et de tableaux vues dans les chapitres 3 et 4.

Exercice TP4.1

1. Écrire une fonction `Somme(L)` qui, pour une liste de flottants `L` renvoie la somme de ses termes (on n'utilisera pas `sum`).
2. Écrire une fonction `Somme_suivant(a,n)` qui, pour un flottant `a` et un entier `n` renvoie une liste de longueur `n` dont la premier terme est `a` et où chaque terme est la somme de tous les termes précédents. Par exemple `Somme_suivant(2,5)` renvoie `[2,2,4,8,16]`. On pourra d'abord utiliser la question précédente puis réfléchir à une optimisation du code.

Exercice TP4.2

Dans cet exercice, on vous demande d'implémenter les fonctions suivantes sur les listes (pour chaque fonction, la liste originale ne doit pas être modifiée).

1. Écrire une fonction `Appartient` qui prend en entrée une liste `L` et un objet `x` et qui renvoie `True` si `x` est dans la liste `L` et `False` sinon.
2. Écrire une fonction `Inserer` qui prend en entrée une liste `L`, un objet `x` et un entier naturel `k` et qui renvoie la liste obtenue en insérant `x` à l'indice `k` dans la liste `L`. Si `k` est supérieur ou égal à la longueur de `L`, on ajoutera `x` à la fin de la liste `L`.
3. Écrire une fonction `Supprimer` qui prend en entrée une liste `L` et un entier naturel `k` et qui renvoie la liste obtenue en supprimant l'objet d'indice `k` de la liste `L`. Si `k` est supérieur ou égal à la longueur de `L`, on renverra un message d'erreur.
4. Écrire une fonction `Indice` qui prend en entrée une liste `L` et un objet `x` et qui renvoie l'indice du premier élément de `L` égal à `x`. On renverra la valeur `-1`, si `x` n'apparaît pas dans la liste `L`.

Exercice TP4.3

Dans cet exercice, on vous demande d'implémenter une fonction sur les tableaux à deux dimensions de type `ndarray`. On rappelle que si x_1, \dots, x_n sont des nombres réels, on peut définir leur moyenne m et leur écart-type σ par

$$m = \frac{1}{n} \sum_{k=1}^n x_k \quad \text{et} \quad \sigma = \sqrt{\frac{1}{n} \sum_{k=1}^n (x_k - m)^2}.$$

Écrire une fonction `EcartType` qui prend en entrée un tableau puis renvoie l'écart type des éléments du tableau.

Exercice TP4.4

1. Écrire une fonction `AppartientTableau` qui prend en entrée un tableau et un élément puis renvoie `True` si l'élément est dans le tableau et `False` sinon.
2. Écrire deux fonctions `MaxTableau` et `indiceMaxTableau` qui prennent chacune en entrée un tableau puis renvoient respectivement le plus grand terme du tableau et le tuple de ces indices.
3. Écrire deux fonctions `FusionLignes` et `FusionColonnes` qui prennent chacune en entrée deux tableaux ayant même nombre de lignes (resp. de colonnes) et qui renvoient respectivement le tableau obtenu en fusionnant leurs lignes (resp. leurs colonnes).

On pourra utiliser la fonction `list`, qui permet de transformer un tuple, ou un tableau contenant une seule ligne en une liste.

Exercice TP4.5

Soit $L = [x_0, \dots, x_{n-1}]$ une liste. On rappelle qu'une sous-liste A de L est une liste de la forme $A = L[i : j] = [x_i, \dots, x_{j-1}]$, avec $0 \leq i \leq j \leq n$. Par exemple, la liste `[1, 2]` est une sous-liste de `[5, 7, 1, 2, 3, 4]`, mais pas de `[3, 4, 1, 5, 2]`.

1. Écrire une fonction `Estsousliste` qui prend en entrée deux listes A et L et qui renvoie `True` si A est une sous-liste de L , `False` dans le cas contraire. Pour cela, on pourra énumérer toutes les sous-listes de L de même longueur que A .
2. Écrire une fonction `NombresousListe` qui compte le nombre de sous-listes de L égales à A .
3. Les deux fonctions précédentes fonctionnent-elles encore si A et L sont des chaînes de caractères? Si non, modifiez-les pour que ce soit le cas.

Exercice TP4.6

Dans cet exercice, un *ensemble* désignera une liste dont les éléments sont deux à deux distincts. On dira que deux ensembles sont égaux s'ils ont les mêmes éléments. Attention, un même ensemble peut donc être représenté par plusieurs listes. Par exemple, les ensembles $[1, 3, 2]$ et $[3, 1, 2]$ sont les mêmes. On pourra utiliser dans cet exercice la fonction **Appartient** de l'exercice 1.

1. Écrire une fonction **Ensemble** qui prend en entrée une liste (dont les éléments ne sont pas nécessairement deux à deux distincts) et qui renvoie l'ensemble des éléments de cette liste.
2. Écrire une fonction **Inclus** qui prend en entrée deux ensembles A et B et qui renvoie le booléen **True** si A est inclus dans B , et **False** dans le cas contraire. En déduire une fonction **Egaux** qui teste l'égalité de deux ensembles.
3. Écrire les fonctions **Reunion**, **Intersection** et **Complement** qui prennent en entrée deux ensembles A et B et qui renvoient respectivement leur réunion, leur intersection et le complémentaire de A dans B .
4. Écrire une fonction **Mots** qui prend en entrée une chaîne de caractères et qui renvoie l'ensemble de ses mots (ici, un mot sera un ensemble de caractères délimité par des espaces ou symboles de ponctuation). Utiliser cette fonction pour compter le nombre de mots distincts d'un texte.