
TP 1 - Listes et tableaux

1 Minimum et du maximum d'une liste

On rappelle qu'on définit une liste `L` en Python grâce à la commande `L=[1,2,4,5]` par exemple. On accède à la composante numéro i de la liste (attention : on commence à compter à 0) par la commande `L[i]`. Par exemple, ici, `L[1]=2`.

1. Écrire une fonction `Maximum` qui prend en paramètre une liste de réels `L` et qui renvoie son maximum (sans utiliser la fonction `max` de Python).
2. Faire de même une fonction `Minimum`.
3. Écrire une fonction `PositionMax` qui renvoie la position du maximum dans la liste. S'il apparaît plusieurs fois, le programme doit renvoyer la liste des positions. Par exemple, pour `L=[1,2,0,1,2]`, on doit obtenir `[1,4]`.

2 Statistiques d'une liste

4. Écrire une fonction `Occurence(L,x)` qui renvoie le nombre d'occurrences (*le nombre d'apparitions*) de `x` dans la liste `L`.
5. On dispose d'une liste `L` regroupant les années de naissance d'un groupe de personnes. Créer une fonction `ClasseEffectif` qui prend en entrée la liste `L` et qui renvoie deux listes : la liste des classes (liste des années de naissance présentes, où elles n'apparaissent qu'une fois) et la liste des effectifs (liste du nombre de personnes du groupe nées cette année). Par exemple, si `L=[2004,2004,2003,2005,2004,2003,2004]`, la commande `ClasseEffectif(L)` doit renvoyer `[2004,2003,2005] [4,2,1]`.
6. Créer une fonction `Moyenne` qui prend en entrée une liste de réels `L` et renvoie la moyenne des valeurs de la liste.
7. L'utiliser pour créer une fonction `Variance` qui prend en entrée `L` et renvoie la variance des valeurs de la liste.

3 Fonctions usuelles sur les listes

8. Reprogrammer la fonction `index`. En Python, la commande `L.index(x)` renvoie l'indice de la première occurrence de `x` dans la liste `L`. Si `x` n'est pas présent dans la liste `L`, la fonction renvoie `x not in the list`. On souhaite obtenir le même résultat en entrant la commande `indexbis(L,x)`.
9. Reprogrammer la fonction `sum`. En Python, la commande `L.sum()` renvoie la somme des éléments de la liste `L`. On souhaite obtenir le même résultat en entrant la commande `sumbis(L)`.
10. Reprogrammer la fonction `remove`. En Python, la commande `L.remove(x)` modifie la liste `L` en lui retirant la première occurrence de l'élément `x`. On souhaite que la commande `removebis(L,x)` nous renvoie la liste ainsi obtenue.
11. Reprogrammer la fonction `reverse`. En Python, la commande `L.reverse()` modifie la liste `L` en inversant l'ordre de ses composantes. On souhaite que la commande `reversebis(L)` nous renvoie la liste ainsi obtenue.
12. Reprogrammer la fonction `choice` du module `random`. En Python, la commande `rd.choice(L)` (*si on a importé le module `random` comme `rd`*) renvoie un nombre au hasard choisi parmi les éléments de la liste `L`. On souhaite obtenir le même résultat en entrant la commande `choicebis(L)`.

4 Notion de parité

On rappelle que si a et b sont des entiers naturels (avec b non nul), la commande $a\%b$ permet d'obtenir le reste de a dans la division euclidienne par b . Par exemple, $7\%2$ vaut 1 (puisque $7 = 2 \times 3 + 1$).

13. Écrire une fonction booléenne `NbPair` qui prend en entrée une liste de réels L et qui détermine si le nombre composantes de cette liste est pair.
14. Écrire une fonction booléenne `EstPair` qui prend en entrée une liste d'entiers et qui détermine si tous les éléments de cette liste sont pairs.
15. Écrire une fonction `Creer` qui prend en entrée un entier n et qui renvoie la liste de longueur $2n$ dont les termes d'indices pairs sont des 1 et les autres une suite arithmétique de raison 2 et de premier terme 1. Par exemple, `Creer(3)` renvoie `[1,1,1,3,1,5]`.

5 Produit scalaire de deux vecteurs de $\mathbb{R}^n$

On rappelle que si $u = (x_1, \dots, x_n)$ et $v = (y_1, \dots, y_n)$ sont deux vecteurs de $\mathbb{R}^n$ le produit scalaire de u et v est donné par

$$u.v = \sum_{i=1}^n x_i y_i.$$

16. Créer une fonction `ProdScalaire` qui prend en entrée deux listes de taille n de réels représentant les coordonnées de deux vecteurs u et v de $\mathbb{R}^n$ et qui renvoie leur produit scalaire.

6 Tableaux

En Python, un tableau est codé comme une liste de liste. Par exemple, le tableau suivant correspond à la liste de listes L :

1	2	3	4
-1	-2	-3	-4
2	6	4	8

$L = [[1, 2, 3, 4], [-1, -2, -3, -4], [2, 6, 4, 8]]$

Les actions fondamentales sont les suivantes :

- on accède au coefficient situé en ligne i et colonne j par la syntaxe `L[i][j]`.
 - on accède au nombre de lignes du tableau par `len(L)`
 - on accède au nombre de colonnes par `len(L[0])`.
17. Écrire une fonction `EstPresent(L,x)` qui renvoie un booléen selon que x est présent ou non dans le tableau L .
 18. (a) Écrire une fonction `MoyenneLigne(L)` qui renvoie la liste contenant les moyennes de chaque ligne du tableau L .
(b) Écrire une fonction `MoyenneColonne(L)` qui renvoie la liste contenant les moyennes de chaque colonne du tableau L .
 19. Écrire une fonction `Min(L)` qui renvoie le minimum du tableau L .
 20. Écrire une fonction `PositionMin(L)` qui renvoie la position du minimum dans le tableau L .
 21. Écrire une fonction `Frequence(L)` qui renvoie la liste des fréquences d'apparition de chacun des entiers compris entre -4 et 4 dans L . Sur le tableau donné en exemple, on attend en sortie `[0.083, 0.083, 0.083, 0.083, 0.0, 0.083, 0.17, 0.083, 0.17]`.
 22. Écrire une fonction `LignePaire(L)` qui renvoie une liste de booléens précisant si les lignes du tableau L sont constituées uniquement d'entiers pairs ou non. Sur le tableau donné en exemple, on attend en sortie `[False, False, True]`.
 23. Écrire une fonction `Produit(L1,L2)` qui à partir de deux listes renvoie le tableau constitué des produits des éléments de $L2$ par ceux de $L1$. Par exemple, `Produit([2,3],[4,5,6])` doit renvoyer `[[8,10,12],[12,15,18]]`.