

On vous rappelle qu'il est nécessaire de tester chaque fonction pour savoir si elle répond à ce qui est demandé. Pour chaque fonction, on réfléchira quelles valeurs sont pertinentes pour juger de son comportement.

Exercice 1

Ecrivez une fonction qui prend en entrée une liste de nombres et renvoie la sous-liste de ses éléments compris entre -1 et 1.

Par exemple si $x = [0.2, -0.5, 2, -0.2, -1.3]$, elle renvoie $[0.2, -0.5, -0.2]$

Exercice 2

Produisez puis affichez la liste des valeurs $4x+z$ où x et z varient chacun entre 0 inclus et 5 exclu.

Le résultat attendu est donc (pas forcément dans cet ordre) :
[0, 1, 2, 3, 4, 4, 5, 6, 7, 8, 8, 9, 10, 11, 12, 12, 13, 14, 15, 16, 16, 17, 18, 19, 20]

Exercice 3

Produisez sous forme d'une liste de listes la matrice de taille 5×5 telle que l'intersection de la colonne x et de la ligne z vaut $4x+z$.

Le résultat attendu est donc :
[[0, 1, 2, 3, 4], [4, 5, 6, 7, 8], [8, 9, 10, 11, 12], [12, 13, 14, 15, 16], [16, 17, 18, 19, 20]]

Exercice 4

Ecrivez une fonction qui prend en entrée une liste x et renvoie un dictionnaire indiquant pour chaque élément de x son nombre d'apparitions dans x .

Par exemple si $x = [2, 5, 2, 2, 1]$, elle renvoie $\{1 : 1, 2 : 3, 5 : 1\}$

Exercice 5

Ecrivez une fonction qui prend en entrée un dictionnaire de listes et supprime toutes les entrées n'ont pas une taille de 4.

Par exemple si $x = \{ "a" : [2, 4, 7, 8], "b" : [3, 4, 8], "c" : [2, 5, 5, 8], "d" : [2, 4, 7, 8, 6] \}$, elle renvoie $\{ "a" : [2, 4, 7, 8], "c" : [2, 5, 5, 8] \}$

Exercice 6

Ecrivez une fonction qui prend en entrée un dictionnaire de listes et supprime toutes les entrées qui contiennent des nombres négatifs.

Par exemple si $x = \{ "a" : [2, 4, 7, -8], "b" : [3, -4, 8], "c" : [2, 5, 5, 8], "d" : [2, 4, 7, 8, 6] \}$, elle renvoie $\{ "c" : [2, 5, 5, 8], "d" : [2, 4, 7, 8, 6] \}$

Exercice 7

Vous êtes une petite souris située au point 0 dans un tunnel monodimensionnel. Ce tunnel contient des morceaux de fromage dont les positions sont indiquées par une liste. Programmez votre déplacement de façon à ce qu'à chaque étape :

1. vous vous dirigez vers le fromage le plus proche (mettre à jour votre position)
2. vous le mangez (mettre à jour la liste)

A la fin, votre algorithme doit renvoyer la liste des fromages dans l'ordre où ils sont mangés ainsi que la distance totale parcourue.

Par exemple, si la liste d'entrée est $[-6, -1, 2, 5]$, l'algorithme renvoie $[-1, 2, 5, -6]$ et une distance totale de $1 + 3 + 3 + 11 = 18$.