Fiche 2 bis

# Épreuves pratiques autour des tableaux

Exercice 1 : Programmer la fonction `moyenne` prenant en paramètre un tableau d'entiers tab (type list) qui renvoie la moyenne de ses éléments si le tableau est non vide et affiche "erreur" si le tableau est vide.

**Exemples :**

```
>>> moyenne([5,3,8])
5.333333333333333
>>> moyenne([1,2,3,4,5,6,7,8,9,10])
5.5
>>> moyenne([])
'erreur'
```

Correction

```
def moyenne (tab) :
    s = 0
    if tab ==[] :
        print("erreur")
    else :
        for i in tab :
            s += i
        return s/(len(tab))
```

Exercice 2 : Écrire une fonction `recherche` qui prend en paramètres `caractere`, un caractère, et `mot`, une chaîne de caractères, et qui renvoie le nombre d'occurrences de `caractere` dans `mot`, c'est-à-dire le nombre de fois où `caractere` dans apparaît dans `mot`.

**Exemple :**

```
>>> recherche('e', "sciences")
2
>>> recherche('i',"mississippi")
4
>>> recherche('a',"mississippi")
0
```

Correction

```
def recherche(caractere,mot):
    nb = 0
    for c in mot:
        if c == caractere:
            nb +=1

    return nb

assert recherche('e', "sciences")== 2, "erreur"
assert recherche('i',"mississippi")==4 , "erreur"
assert recherche('a',"mississippi")==0 , "erreur"
```

Exercice 3 : Soit le couple (`note`,`coefficient`) :

- `note` est un nombre de type flottant (`float`) compris entre 0 et 20 ;
- `coefficient` est un nombre entier positif.

Les résultats aux évaluations d'un élève sont regroupés dans une liste composée de couples (`note` ,`coefficient`).
Écrire une fonction `moyenne` qui renvoie la moyenne pondérée de cette liste donnée en paramètre.

Par exemple, l'expression `moyenne`([(15, 2), (9, 1), (12, 3)]) devra renvoyer le résultat du calcul suivant :

$$\frac{2\times 15+1\times 9+3\times 12}{2+1+3}=12,5$$

Correction

```
def moyenne (liste_note) :
    somme_coef = 0
    somme_note = 0
    for (n,c) in liste_note :
        somme_coef += c
        somme_note += n*c
    return somme_note/somme_coef
```

Exercice 4 : Écrire une fonction `maxi` qui prend en paramètre une liste `tab` de nombres entiers et renvoie un couple donnant le plus grand élément de cette liste, ainsi que l'indice de la première apparition de ce maximum dans la liste.

**Exemple :**

```
>>> maxi([1,5,6,9,1,2,3,7,9,8])
(9,3)
```

Correction

```
def maxi(tab):
    maxi_tab = tab[0]
    indice_max = 0
    for i in range(1,len(tab)) :
        if tab[i]> maxi_tab :
            maxi_tab = tab[i]
            indice_max =i
    return (maxi_tab , indice_max)
```

Exercice 5 : Écrire une fonction `recherche` qui prend en paramètres `elt` un nombre et `tab` un tableau de nombres, et qui renvoie le tableau des indices de `elt` dans `tab` si `elt` est dans `tab` et le tableau vide [] sinon.

**Exemple :**

```
>>> recherche(3, [3, 2, 1, 3, 2, 1])
[0, 3]
>>> recherche(4, [1, 2, 3])
[]
```

Correction

```
def recherche(elt,tab):
    result = []
    for i in range(len(tab)):
        if tab[i] == elt :
            result.append(i)
    return result
```

Exercice 6 : Écrire une fonction `rechercheMinMax` qui prend en paramètre un tableau de nombres non triés `tab`, et qui renvoie la plus petite et la plus grande valeur du tableau sous la forme d'un dictionnaire à deux clés `'min'` et `'max'`. Les tableaux seront représentés sous forme de liste Python.

**Exemple :**

```
>>> tableau = [0, 1, 4, 2, -2, 9, 3, 1, 7, 1]
>>> resultat = rechercheMinMax(tableau)
>>> resultat
{'min': -2, 'max': 9}

>>> tableau = []
>>> resultat = rechercheMinMax(tableau)
>>> resultat
{'min': None, 'max': None}
```

Correction

```
def rechercheMinMax(tab):
    if len(tab) == 0 :
        mini = None
        maxi = None
    else :
        mini = tab[0]
        maxi = tab[0]
        for i in tab:
            if i < mini :
                mini = i
            if i > maxi :
                maxi = i
    return {'min' : mini , 'maxi' : maxi}

tableau = [0, 1, 4, 2, -2, 9, 3, 1, 7, 1]
print(rechercheMinMax(tableau))
```

```
tableau = []
print(rechercheMinMax(tableau))
```

Exercice 7 : Écrire une fonction `indice_du_min` qui prend en paramètre un tableau de nombres non trié `tab`, et qui renvoie l'indice de la première occurrence du minimum de ce tableau. Les tableaux seront représentés sous forme de liste Python.

**Exemple :**

```
>>> indice_du_min([5])
0
>>> indice_du_min([2, 4, 1])
2
>>> indice_du_min([5, 3, 2, 2, 4])
2
```

Correction

```
def indice_du_min(tab):
    indice = 0
    mini = tab[0]
    for i in range(len(tab)):
        if tab[i]<mini :
            indice = i
            mini = tab[i]
    return indice
```

Exercice 8 : On a relevé les valeurs moyennes annuelles des températures à Paris pour la période allant de 2013 à 2019. Les résultats ont été récupérés sous la forme de deux listes : l'une pour les températures, l'autre pour les années :

```
t_moy = [14.9, 13.3, 13.1, 12.5, 13.0, 13.6, 13.7]
annees = [2013, 2014, 2015, 2016, 2017, 2018, 2019]
```

Écrire la fonction `mini` qui prend en paramètres le tableau `releve` des relevés et le tableau `date` des dates et qui renvoie la plus petite valeur relevée au cours de la période et l'année correspondante.

**Exemple :**

```
>>> mini(t_moy, annees)
12.5, 2016
```

Correction

```
t_moy = [14.9, 13.3, 13.1, 12.5, 13.0, 13.6, 13.7]
annees = [2013, 2014, 2015, 2016, 2017, 2018, 2019]

def mini(releve,date) :
    note_min = releve[0]
    indice = 0
    for i in range(1,len(releve)) :
        if releve[i] < note_min :
            note_min = releve[i]
            indice = i
    return ( note_min , date[indice])

print(mini(t_moy,annees))
```