

## DS - Première NSI – 1h

### EXERCICE 1 (4 points)

Écrire une fonction `enumere` qui prend en paramètre un tableau `tab` (type list) et renvoie un dictionnaire `d` dont les clés sont les éléments de `tab` avec pour valeur associée la liste des indices de l'élément dans le tableau `tab`.

Exemples :

```
>>> enumere([])
{}
>>> enumere([1, 2, 3])
{1: [0], 2: [1], 3: [2]}
>>> enumere([1, 1, 2, 3, 2, 1])
{1: [0, 1, 5], 2: [2, 4], 3: [3]}
```

### EXERCICE 2 (4 points)

Programmer la fonction `recherche`, prenant en paramètre un tableau non vide `tab` (type list) d'entiers et un entier `n`, et qui renvoie l'indice de la dernière occurrence de l'élément cherché. Si l'élément n'est pas présent, la fonction renvoie `None`.

Exemples :

```
>>> recherche([5, 3], 1)
None
>>> recherche([2, 4], 2)
0
>>> recherche([2, 3, 5, 2, 4], 2)
3
```

### EXERCICE 3 (4 points)

Dans cet exercice on cherche à calculer la moyenne pondérée d'un élève dans une matière donnée. Chaque note est associée à un coefficient qui la pondère.

Par exemple, si ses notes sont : 14 avec coefficient 3, 12 avec coefficient 1 et 16 avec coefficient 2, sa moyenne pondérée sera donnée par

$(14 \times 3 + 12 \times 1 + 16 \times 2) / (3 + 1 + 2) = 14,333\dots$

Écrire une fonction `moyenne` :

- qui prend en paramètre une liste `notes` non vide de tuples à deux éléments entiers de la forme (note, coefficient) (int ou float) positifs ou nuls ;
- et qui renvoie la moyenne pondérée des notes de la liste sous forme de flottant si la somme des coefficients est non nulle, `None` sinon.

Exemple :

```
>>> moyenne([(8, 2), (12, 0), (13.5, 1), (5, 0.5)])
9.142857142857142
>>> moyenne([(3, 0), (5, 0)])
None
```

**EXERCICE 4 (4 points)**

Écrire une fonction `a_doublon` qui prend en paramètre un tableau trié de nombres dans l'ordre croissant et renvoie `True` si ce tableau contient au moins deux nombres identiques, `False` sinon.

Exemple :

```
>>> a_doublon([])
False
>>> a_doublon([1])
False
>>> a_doublon([1, 2, 4, 6, 6])
True
>>> a_doublon([2, 5, 7, 7, 7, 9])
True
>>> a_doublon([0, 2, 3])
False
```

**EXERCICE 5 (4 points)**

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 tableaux (de type `list`) : l'un 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 `annee_temperature_minimale` qui prend en paramètres ces deux tableaux et qui renvoie la plus petite valeur relevée au cours de la période et l'année correspondante.

On suppose que la température minimale est atteinte une seule fois.

Exemple :

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