

## Corrigé sujet **25** - Année : 2023

Sujet 25 - 2023 ⬇

### Exercice 1

```
1 def enumere(L):
2     d = {}
3     for i in range(len(L)):
4         if L[i] in d:
5             d[L[i]].append(i)
6         else:
7             d[L[i]] = [i]
8     return d
```

#### Commentaire

On parcourt la liste par indice, si on a déjà rencontré l'élément alors on met à jour sa liste d'indice sinon on crée une clé dans le dictionnaire dont la valeur est la liste contenant l'indice.

### Exercice 2

```
1 class Arbre:
2     def __init__(self, etiquette):
3         self.v = etiquette
4         self.fg = None
5         self.fd = None
6
7     def parcours(arbre, liste):
8         if arbre != None:
9             parcours(arbre.fg, liste)
10            liste.append(arbre.v)
11            parcours(arbre.fd, liste)
12        return liste
13
14    def insere(arbre, cle):
15        """ arbre est une instance de la classe Arbre qui implémente
16            un arbre binaire de recherche.
17        """
18        if cle < arbre.v: # ①
19            if arbre.fg != None: # ②
20                insere(arbre.fg, cle)
21            else:
22                arbre.fg = Arbre(cle)
```

```
23         arbre.fg = Arbre(cle)
24     else:
25         if arbre.fd != None: # 3
26             insere(arbre.fd, cle)
27         else:
28             arbre.fd = Arbre(cle)
```

1. On teste si `cle` est inférieure à l'étiquette du noeud, dans ce cas il faut insérer à gauche
2. Si le fils gauche est `None` on insère à cet endroit, sinon on insère dans l'arbre gauche
3. Traitement identique pour le côté droit



#### Attention

- La fonction `insere` devrait être une méthode de la classe `ABR`
- Le sujet ne précise pas le comportement à adopter si on insère une clé existante