

## Sujet 2

### ❑ Exercice 1 : *Liste des indices*

Écrire une fonction `liste_indices` qui prend en paramètres un entier `n` et une liste d'entiers `l` et qui renvoie la liste des indices d'apparition de `n` dans `l`. Par exemple, `liste_indices(5, [17, 5, 11, 5, 5, 14])` renvoie `[1, 3, 4]`. Lorsque l'entier `n` n'apparaît pas dans la liste, on renvoie la liste vide.

**Exemples :**

```
1 >>> liste_indice(13, [13, 24, 10, 13])
2 [0, 3]
3 >>> liste_indice(11, [7, 19, 15, 14, 2])
4 []
5 >>> liste_indice(14, [7, 14, 15, 28, 2])
6 [1]
```

### ❑ Exercice 2 : *Recherche dichotomique dans une liste triée*

On rappelle que lorsque qu'une liste est *déjà triée*, on peut y rechercher un élément grâce à la recherche dichotomique (plus performante que la recherche simple). Cet algorithme consiste à rechercher entre deux indices `debut` (initialisé à 0) et `fin` (initialisé à la longueur de la liste moins 1). À chaque étape on compare l'élément recherché avec celui situé au milieu de ces deux indices et on les mets à jour en conséquence. Compléter l'implémentation itérative suivante de la recherche dichotomique :

```
1 def dichotomie(liste, element):
2     ''' recherche element dans liste en utilisant la recherche
3         dichotomique '''
4     debut = ...
5     fin = ....
6     while debut <= fin:
7         milieu = .....
8         if liste[milieu] == element:
9             return .....
10        if liste[milieu] > element:
11            fin = .....
12        else:
13            debut = ....
14    return ....
```

**Exemples :**

```
1 >>> dichotomie([15, 15, 15, 18, 22, 31], 15)
2 True
3 >>> dichotomie([7, 11, 12, 18, 22, 31], 28)
4 False
```