
DIVISER POUR RÉGNER

Exercice 0 Donne la séquence des appels à la fonction `recherche`, rappelée ci-dessous, pendant l'évaluation des deux appels suivants :

- `recherche([0,1,1,2,3,5,8,13,21], 13, 0, 8)`
- `recherche([0,1,1,2,3,5,8,13,21], 12, 0, 8)`

```
1 def recherche(T, v, g, d):
2     m = ( g + d )//2
3     if g > d:
4         | return None
5     elif T[m] == v:
6         | return m
7     elif T[m] < v:
8         | return recherche(T, v, m+1, d)
9     elif T[m] > v:
10        | return recherche(T, v, g, m-1)
```

Exercice 1 Réécris la fonction du TP `fusion(l1, l2)` de manière récursive plutôt qu'avec une boucle `while`.

Indication : On pourra introduire deux nouveaux paramètres dans la définition de la fonction, permettant de sélectionner dans chacune des listes le bon élément à ajouter à la nouvelle liste.

```
1 def fusion(l1, l2):
2     L = []
3     n1, n2 = len(l1), len(l2)
4     i1, i2 = 0, 0
5     while i1 < n1 and i2 < n2:
6         | if l1[i1] <= l2[i2]:
7             | L.append(l1[i1])
8             | i1 += 1
9         | else:
10            | L.append(l2[i2])
11            | i2 += 1
12    for i in range(i1, len(l1)):
13        | L.append(l1[i])
14    for i in range(i2, len(l2)):
15        | L.append(l2[i])
16    return L
```