

SUJET 41

EXERCICE 1 : 3 versions du même code.

```
1 def recherche(caractere:str,chaîne:str)->int:
2     nb_oc = 0
3     for i in range(len(chaîne)):
4         if chaîne[i] == caractere:
5             nb_oc = nb_oc + 1
6     return nb_oc
7
8
9 def recherche2(caractere:str,chaîne:str)->int:
10    nb_oc = 0
11    for elt in chaîne:
12        if elt == caractere:
13            nb_oc = nb_oc + 1
14    return nb_oc
15
16
17 def recherche3(caractere:str,chaîne:str)->int:
18     ''' version recursive '''
19     if caractere not in chaîne:
20         return 0
21     else:
22         if chaîne[0] != caractere:
23             return 0+recherche3(caractere, chaîne[1:])
24         else:
25             return 1+recherche3(caractere, chaîne[1:])
26
```

EXERCICE 2 :

```
1 valeurs = [100, 50, 20, 10, 5, 2, 1]
2
3 def rendu_glouton(a_rendre, rang)->list:
4     if a_rendre == 0:
5         return []
6     v = valeurs[rang]
7     if v <= a_rendre :
8         return [v] + rendu_glouton(a_rendre - v, rang)
9     else :
10        return rendu_glouton(a_rendre, rang+1)
```

Ligne 5 :

Ici on est dans le cas trivial (ou cas de base). S'il n'y a pas d'argent à rendre, comme on doit renvoyer une liste de valeurs, on renvoie une liste vide

Ligne 7 :

On va étudier les 2 cas suivant si la valeur pièce regardée est supérieure ou inférieure à la somme à rendre.

Ligne 8 :

Si la valeur de la pièce est inférieur à la somme à rendre. On ajoute v à la liste réponse **[v] +** et on regarde comment rendre la monnaie sur la nouvelle somme à rendre **(a_rendre-v)**

Ligne 10 :

Si la valeur de la pièce est supérieur à la somme à rendre. On ne peut pas rendre cette pièce. On regarde donc comment on fait pour rendre la monnaie avec les pièces suivantes (le rang augmente de 1)