

Corrigé sujet 01 - Année : 2022

[Sujet 01 - 2022 ↴](#)

1. Exercice 1



```

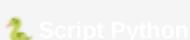
1 def recherche(caractere,mot):
2     occurrence = 0
3     for c in mot:
4         if c == caractere:
5             occurrence += 1
6     return occurrence

```

Commentaires

C'est un exercice classique de parcours d'un itérable en comptant les occurrences d'apparition d'une valeur. Un parcours par élément suffit, les indices des occurrences n'étant pas utilisées.

2. Exercice 2



```

1 Pièces = [100,50,20,10,5,2,1]
2 def rendu_glouton(arendre, solution=[], i=0):
3     if arendre == 0:
4         return solution #(1)
5     p = Pièces[i]
6     if p <= arendre : #(2)
7         solution.append(p)
8         return rendu_glouton(arendre - p, solution, i)
9     else :
10        return rendu_glouton(arendre, solution, i+1) #(3)

```

1. Lorsqu'il n'y a plus rien à rendre on renvoie la solution
2. Si la pièce est plus petite que la somme à rendre on l'ajoute à la solution et on diminue la somme à rendre
3. La pièce dépasse la somme à rendre, on relance le processus en regardant la pièce suivante dans la liste

 **Attention**

- Le code fourni utilise un objet mutable (une liste) comme paramètre par défaut d'une fonction :

 **Script Python**

```
def rendu_glouton(arendre, solution=[], i=0):
```

C'est une très mauvaise pratique car source d'erreurs, en effet la variable `solution` étant mutable elle est modifiée par la fonction lors d'un premier appel et ne sera donc plus vide lors des appels suivants. Pour constater le problème, faire plusieurs appels à cette fonction sans spécifier les arguments ayant des valeurs par défaut. Pour une solution à ce problème, on pourra par exemple consulter [ce site](#)