

# NUMERIQUE et SCIENCES INFORMATIQUES

Épreuve de l'enseignement de spécialité

Sujet d'entraînement

Partie Pratique

Classe Terminale de la voie générale

Le candidat doit traiter les 2 exercices

Ce sujet comporte 3 pages



## Exercice 1

Écrire une fonction **supprime** qui prend en paramètres **val**, une valeur, et **lst** une liste, qui affiche le nombre total d'éléments correspondant à **val** supprimés, et qui renvoie une liste dépourvue de ces éléments.

Par exemple, `supprime(2, [2, 4, 2, 5, 6, 2, 7, 4, 5, 7, 9])` affiche :

3 éléments ont été supprimés.

et renvoie :

[4, 5, 6, 7, 4, 5, 7, 9]

## Exercice 2

Algorithme du rendu de monnaie avec un certain nombre de pièces/billets disponibles.

On ne disposera que des pièces et billets suivants :

---

|   |     |     |     |    |    |    |   |   |   |
|---|-----|-----|-----|----|----|----|---|---|---|
| Valeur de la pièce ou du billet (en centimes) | 500 | 200 | 100 | 50 | 20 | 10 | 5 | 2 | 1 |
| Nombre de pièces ou de billets correspondants | 0   | 0   | 0   | 0  | 1  | 2  | 1 | 3 | 1 |

---

---

|   |       |       |       |      |      |      |
|---|-------|-------|-------|------|------|------|
| Valeur de la pièce ou du billet (en centimes) | 50cts | 20cts | 10cts | 5cts | 2cts | 1cts |
| Nombre de pièces ou de billets correspondants | 1     | 3     | 0     | 1    | 0    | 2    |

---

La stratégie gloutonne est utilisée : on essaie de rendre la monnaie en maximisant le nombre de pièces de grande valeur.

Exemple : s'il faut rendre 46 centimes, on prend autant de pièces de 20cts que possible sans dépasser la somme à rendre, puis autant de pièces de 10cts que possible sans dépasser le reste de la somme à rendre, etc.

La fonction **rendre\_monnaie** prend en paramètres **somme\_a\_rendre**, un nombre entier correspondant à la somme à rendre (exprimée en centimes), et de **pieces\_dispo**, un dictionnaire associant la valeur faciale des pièces (ou billets) et renvoie la liste des pièces (ou billets) à rendre :

```
>>>pieces_disponibles={
    500_00:0,
    200_00:0,
    100_00:0,
    50_00:0,
    20_00:1,
    10_00:2,
    5_00:1,
    2_00:3,
    1_00:1,
    50:1,
```

```
    20:3,  
    10:0,  
    5:1,  
    2:0,  
    1:2  
}  
>>> rendre_monnaie(46, pieces_disponibles)  
[20, 20, 5, 1]
```

Compléter le code Python ci-dessous pour qu'il implémente la fonction `rendre_monnaie`.

```
def rendre_monnaie(somme_a_rendre, pieces_dispo):  
    # liste des pièces à rendre  
    lst_a_rendre = []  
    # boucle de construction de la liste des pièces  
    for elt in pieces_dispo :  
        while ... and ... and ...:  
            somme_a_rendre = ...  
            pieces_dispo[elt]-=1  
            lst_a_rendre.append(...)  
    return lst_a_rendre
```