

# **BACCALAURÉAT**

**SESSION 2023**

---

**Épreuve de l'enseignement de spécialité**

## **NUMÉRIQUE et SCIENCES INFORMATIQUES**

**Partie pratique**

**Classe Terminale de la voie générale**

---

**Sujet n°13**

---

**DURÉE DE L'ÉPREUVE : 1 heure**

**Le sujet comporte 3 pages numérotées de 1 / 3 à 3 / 3  
Dès que le sujet vous est remis, assurez-vous qu'il est complet.**

*Le candidat doit traiter les 2 exercices.*

## EXERCICE 1 (4 points)

Écrire en langage Python une fonction `recherche` prenant comme paramètres une variable `a` de type numérique (`float` ou `int`) et un tableau `tab` (de type `list`) et qui renvoie le nombre d'occurrences de `a` dans `tab`.

Exemples d'utilisations de la fonction `recherche` :

```
>>> recherche(5, [])
0
>>> recherche(5, [-2, 3, 4, 8])
0
>>> recherche(5, [-2, 3, 1, 5, 3, 7, 4])
1
>>> recherche(5, [-2, 5, 3, 5, 4, 5])
3
```

## EXERCICE 2 (4 points)

La fonction `rendu_monnaie` prend en paramètres deux nombres entiers positifs `somme_due` et `somme_versee`. Elle procède au rendu de la monnaie de la différence `somme_versee - somme_due` pour des achats effectués avec le système monétaire de la zone Euro. On utilise pour cela un algorithme glouton qui commence par rendre le maximum de pièces ou billets de plus grandes valeurs et ainsi de suite. Par la suite, on assimilera les billets à des pièces.

La fonction `rendu_monnaie` renvoie un tableau de type `list` contenant les pièces qui composent le rendu.

Toutes les sommes sont exprimées en euros. Les valeurs possibles pour les pièces sont donc contenues dans le tableau `pieces = [1, 2, 5, 10, 20, 50, 100, 200]`.

Ainsi, l'instruction `rendu_monnaie(452, 500)` renvoie le tableau `[20, 20, 5, 2, 1]`. En effet, la somme à rendre est de 48 euros soit  $20 + 20 + 5 + 2 + 1$ .

Le code de la fonction `rendu_monnaie` est donné ci-dessous :

```
def rendu_monnaie(somme_due, somme_versee):
    rendu = ...
    a_rendre = ...
    i = len(pieces) - 1
    while ... :
        if pieces[i] <= a_rendre:
            rendu.append(...)
            a_rendre = ...
        else:
            i = ...
    return rendu
```

Compléter ce code et le tester.

Exemples :

```
>>> rendu_monnaie(700, 700)
[]
```

```
>>> rendu_monnaie(102, 500)
[200, 100, 50, 20, 20, 5, 2, 1]
```