

# BACCALAURÉAT

SESSION 2025

---

Épreuve de l'enseignement de spécialité

## NUMÉRIQUE et SCIENCES INFORMATIQUES

Partie pratique

Classe Terminale de la voie générale

---

Sujet n°32

---

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 (10 points)

Écrire une fonction `occurrences(caractere, chaîne)` qui prend en paramètres `caractere`, une chaîne de caractère de longueur 1, et `chaîne`, une chaîne de caractères.

Cette fonction renvoie le nombre d'occurrences de `caractere` dans `chaîne`, c'est-à-dire le nombre de fois où `caractere` apparaît dans `chaîne`.

Exemples :

```
>>> occurrences('e', "sciences")
2
>>> occurrences('i', "mississippi")
4
>>> occurrences('a', "mississippi")
0
```

## EXERCICE 2 (10 points)

On s'intéresse à un algorithme récursif qui permet de rendre la monnaie à partir d'une liste donnée de valeurs de pièces et de billets.

Le système monétaire est donné sous forme d'une liste `valeurs = [100, 50, 20, 10, 5, 2, 1]`. On suppose que les pièces et les billets sont disponibles sans limitation.

On cherche à donner la liste des valeurs à rendre pour une somme donnée en argument. L'algorithme utilisé est de type glouton.

Compléter le code Python ci-dessous de la fonction `rendu_glouton` qui implémente cet algorithme et renvoie la liste des pièces à rendre.

```
valeurs = [100, 50, 20, 10, 5, 2, 1]
```

```
def rendu_glouton(a_rendre, rang):  
    if a_rendre == 0:  
        return ...  
    v = valeurs[rang]  
    if v <= ...:  
        return ... + rendu_glouton(a_rendre - v, rang)  
    else:  
        return rendu_glouton(a_rendre, ...)
```

On devra obtenir :

```
>>> rendu_glouton(67, 0)  
[50, 10, 5, 2]  
>>> rendu_glouton(291, 0)  
[100, 100, 50, 20, 20, 1]  
>>> rendu_glouton(291,1) # si on ne dispose pas de billets de 100  
[50, 50, 50, 50, 50, 20, 20, 1]
```