

# BACCALAURÉAT

SESSION 2024

---

Épreuve de l'enseignement de spécialité

## NUMÉRIQUE et SCIENCES INFORMATIQUES

Partie pratique

Classe Terminale de la voie générale

---

Sujet n°09

---

DURÉE DE L'ÉPREUVE : 1 heure

**Le sujet comporte 4 pages numérotées de 1 / 4 à 4 / 4  
Dès que le sujet vous est remis, assurez-vous qu'il est complet.**

*Le candidat doit traiter les 2 exercices.*

## EXERCICE 1 (10 points)

On veut trier par ordre croissant les notes d'une évaluation qui sont des nombres entiers compris entre 0 et 10 (inclus).

Ces notes sont contenues dans un tableau `notes_eval` (type `list`).

Écrire une fonction `effectif_notes` prenant en paramètre le tableau `notes_eval` et renvoyant un tableau de longueur 11 tel que la valeur d'indice `i` soit le nombre de notes valant `i` dans le tableau `notes_eval`.

Écrire ensuite une fonction `notes_triees` prenant en paramètre le tableau des effectifs des notes et renvoyant un tableau contenant les mêmes valeurs que `notes_eval` mais triées dans l'ordre croissant.

Exemple :

```
>>> notes_eval = [2, 0, 5, 9, 6, 9, 10, 5, 7,
                  9, 9, 5, 0, 9, 6, 5, 4]

>>> eff = effectif_notes(notes_eval)
>>> eff
[2, 0, 1, 0, 1, 4, 2, 1, 0, 5, 1]

>>> notes_triees(eff)
[0, 0, 2, 4, 5, 5, 5, 5, 6, 6, 7, 9, 9, 9, 9, 9, 10]
```

## EXERCICE 2 (10 points)

L'objectif de cet exercice est d'écrire deux fonctions récursives `dec_to_bin` et `bin_to_dec` assurant respectivement la conversion de l'écriture décimale d'un nombre entier vers son écriture en binaire et, réciproquement, la conversion de l'écriture en binaire d'un nombre vers son écriture décimale.

Dans cet exercice, on s'interdit l'usage des fonctions Python `bin` et `int`.

L'exemple suivante montre comment obtenir l'écriture en binaire du nombre 25 :

$$\begin{aligned} 25 &= 2 \times 12 + 1 \\ &= 2 \times (2 \times 6 + 0) + 1 \\ &= 2 \times (2 \times (2 \times 3 + 0) + 0) + 1 \\ &= 2 \times (2 \times (2 \times (2 \times 1 + 1) + 0) + 0) + 1 \\ &= 2 \times (2 \times (2 \times (2 \times (2 \times 0 + 1) + 1) + 0) + 0) + 1 \\ &= 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\ &= \underline{11001}_2 \end{aligned}$$

L'écriture binaire de 25 est donc 11001.

On rappelle également que

- l'expression `a // 2` calcule le quotient de la division euclidienne de `a` par 2 ;
- l'expression `a % 2` calcule le reste dans la division euclidienne de `a` par 2.

On indique enfin qu'en Python si `mot = "informatique"`, alors

- l'expression `mot[-1]` vaut `'e'`, c'est-à-dire le dernier caractère de la chaîne de caractères `mot` ;
- l'expression `mot[:-1]` vaut `'informatiqu'`, c'est-à-dire l'ensemble de la chaîne de caractères `mot` privée de son dernier caractère.

Compléter, puis tester, le code des deux fonctions situées à la page suivante.

On précise que la fonction récursive `dec_to_bin` prend en paramètre un nombre entier et renvoie une chaîne de caractères contenant l'écriture en binaire du nombre passé en paramètre.

Exemple :

```
>>> dec_to_bin(25)
'11001'
```

La fonction récursive `bin_to_dec` prend en paramètre une chaîne de caractères représentant l'écriture d'un nombre en binaire et renvoie l'écriture décimale de ce nombre.

```
>>> bin_to_dec('101010')
42
```

```

def dec_to_bin(nb_dec):
    q, r = nb_dec // 2, nb_dec % 2
    if q == ...:
        return ...
    else:
        return dec_to_bin(...) + ...

def bin_to_dec(nb_bin):
    if len(nb_bin) == 1:
        if ... == '0':
            return 0
        else:
            return ...
    else:
        if nb_bin[-1] == '0':
            bit_droit = 0
        else:
            ...
        return ... * bin_to_dec(nb_bin[:-1]) + ...

```