

# **BACCALAURÉAT**

**SESSION 2023**

---

**Épreuve de l'enseignement de spécialité**

## **NUMÉRIQUE et SCIENCES INFORMATIQUES**

**Partie pratique**

**Classe Terminale de la voie générale**

---

**Sujet n°45**

---

**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 (4 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 une liste `notes_eval`.

Écrire une fonction `rangement_valeurs` prenant en paramètre la liste `notes_eval` et renvoyant une liste de longueur 11 telle que la valeur de cette liste à chaque rang est égale au nombre de notes valant ce rang. Ainsi le terme de rang 0 indique le nombre de note 0, le terme de rang 1 le nombre de note 1, etc.

Écrire ensuite une fonction `notes_triees` prenant en paramètre la liste des effectifs des notes et renvoyant une liste contenant la liste, triée dans l'ordre croissant, des notes des élèves.

Exemple :

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

```
>>> effectifs_notes = rangement_valeurs(notes_eval)
```

```
>>> effectifs_notes  
[2, 0, 1, 0, 1, 4, 2, 1, 0, 5, 1]
```

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

## EXERCICE 2 (4 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`.

On rappelle sur l'exemple ci-dessous une façon d'obtenir l'écriture en binaire du nombre 25 :

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

L'écriture en binaire de 25 est donc 11001.

On rappelle également que :

`a // 2` renvoie le quotient de la division euclidienne de `a` par 2.

`a % 2` renvoie le reste dans la division euclidienne de `a` par 2.

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

`mot[-1]` renvoie 'e', c'est-à-dire le dernier caractère de la chaîne de caractères `mot`.

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

Compléter, puis tester, les codes de deux fonctions de 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 str(r)
    else:
        return dec_to_bin(...) + ...

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

```