

# **BACCALAUREAT**

**SESSION 2022**

---

**Épreuve de l'enseignement de spécialité**

## **NUMERIQUE et SCIENCES INFORMATIQUES**

**Partie pratique**

**Classe Terminale de la voie générale**

---

**Sujet n°20**

---

DUREE DE L'ÉPREUVE : 1 heure

**Le sujet comporte 2 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)

L'opérateur « ou exclusif » entre deux bits renvoie 0 si les deux bits sont égaux et 1 s'ils sont différents :  $0 \oplus 0 = 0$ ,  $0 \oplus 1 = 1$ ,  $1 \oplus 0 = 1$ ,  $1 \oplus 1 = 0$

On représente ici une suite de bits par un tableau contenant des 0 et des 1.

Exemples :

a = [1, 0, 1, 0, 1, 1, 0, 1]

b = [0, 1, 1, 1, 0, 1, 0, 0]

c = [1, 1, 0, 1]

d = [0, 0, 1, 1]

Écrire la fonction xor qui prend en paramètres deux tableaux de même longueur et qui renvoie un tableau où l'élément situé à position `i` est le résultat, par l'opérateur « ou exclusif », des éléments à la position `i` des tableaux passés en paramètres.

En considérant les quatre exemples ci-dessus, cette fonction doit passer les tests suivants :

```
assert(xor(a, b) == [1, 1, 0, 1, 1, 0, 0, 1])
```

```
assert(xor(c, d) == [1, 1, 1, 0])
```

### EXERCICE 2 (4 points)

Dans cet exercice, on appelle carré d'ordre  $n$  un tableau de  $n$  lignes et  $n$  colonnes dont chaque case contient un entier naturel.

Exemples :

|   |   |
|---|---|
| 1 | 1 |
| 1 | 1 |

c2

|   |   |   |
|---|---|---|
| 2 | 9 | 4 |
| 7 | 5 | 3 |
| 6 | 1 | 8 |

c3

|    |    |    |    |
|----|----|----|----|
| 4  | 5  | 16 | 9  |
| 14 | 7  | 2  | 11 |
| 3  | 10 | 15 | 6  |
| 13 | 12 | 8  | 1  |

c4

Un carré d'ordre 2

Un carré d'ordre 3

Un carré d'ordre 4

Un carré est dit magique lorsque les sommes des éléments situés sur chaque ligne, chaque colonne et chaque diagonale sont égales. Ainsi c2 et c3 sont magiques car la somme de chaque ligne, chaque colonne et chaque diagonale est égale à 2 pour c2 et 15 pour c3. c4 n'est pas magique car la somme de la première ligne est égale à 34 alors que celle de la dernière colonne est égale à 27.

La classe Carre ci-après contient des méthodes qui permettent de manipuler des carrés.

Compléter la fonction `est_magique` qui prend en paramètre un carré et qui renvoie la valeur de la somme si ce carré est magique, False sinon.

```

class Carre:
    def __init__(self, tableau = [[]]):
        self.ordre = len(tableau)
        self.valeurs = tableau

    def affiche(self):
        '''Affiche un carré'''
        for i in range(self.ordre):
            print(self.valeurs[i])

    def somme_ligne(self, i):
        '''Calcule la somme des valeurs de la ligne i'''
        return sum(self.valeurs[i])

    def somme_col(self, j):
        '''Calcule la somme des valeurs de la colonne j'''
        return sum([self.valeurs[i][j] for i in range(self.ordre)])

def est_magique(carre):
    n = carre.ordre
    s = carre.somme_ligne(0)

    #test de la somme de chaque ligne
    for i in range(..., ...):
        if carre.somme_ligne(i) != s:
            return ...

    #test de la somme de chaque colonne
    for j in range(n):
        if ... != s:
            return False

    #test de la somme de chaque diagonale
    if sum([carre.valeurs[...][...] for k in range(n)]) != s:
        return False
    if sum([carre.valeurs[k][n-1-k] for k in range(n)]) != s:
        return False

    return ...

```

Tester la fonction `est_magique` sur les carrés `c2`, `c3` et `c4`.