

# **BACCALAUREAT**

**SESSION 2021**

---

**Épreuve de l'enseignement de spécialité**

## **NUMERIQUE et SCIENCES INFORMATIQUES**

**Partie pratique**

**Classe Terminale de la voie générale**

---

**Sujet n°2**

---

DUREE 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)

Programmer la fonction `moyenne` prenant en paramètre un tableau d'entiers `tab` (type `list`) qui renvoie la moyenne de ses éléments si le tableau est non vide et affiche `'erreur'` si le tableau est vide.

Exemples :

```
>>> moyenne([5, 3, 8])
5.333333333333333
>>> moyenne([1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
5.5
>>> moyenne([])
'erreur'
```

## EXERCICE 2 (4 points)

On considère un tableau d'entiers `tab` (type `list` dont les éléments sont des 0 ou des 1). On se propose de trier ce tableau selon l'algorithme suivant : à chaque étape du tri, le tableau est constitué de trois zones consécutives, la première ne contenant que des 0, la seconde n'étant pas triée et la dernière ne contenant que des 1.

Zone de 0	Zone non triée	Zone de 1
-----------	----------------	-----------

Tant que la zone non triée n'est pas réduite à un seul élément, on regarde son premier élément :

- si cet élément vaut 0, on considère qu'il appartient désormais à la zone ne contenant que des 0 ;
- si cet élément vaut 1, il est échangé avec le dernier élément de la zone non triée et on considère alors qu'il appartient à la zone ne contenant que des 1.

Dans tous les cas, la longueur de la zone non triée diminue de 1.

Recopier sous Python en la complétant la fonction `tri` suivante :

```
def tri(tab):
    #i est le premier indice de la zone non trie, j le dernier
    indice.
    #Au debut, la zone non trie est le tableau entier.
    i= ...
    j= ...
    while i != j :
        if tab[i]== 0:
            i= ...
        else :
            valeur = tab[j]
            tab[j] = ...
            ...
            j= ...
    ...
```

**Exemple :**

```
>>> tri([0,1,0,1,0,1,0,1,0])
[0, 0, 0, 0, 0, 1, 1, 1, 1]
```