

# **BACCALAUREAT**

**SESSION 2022**

---

**Épreuve de l'enseignement de spécialité**

## **NUMERIQUE et SCIENCES INFORMATIQUES**

**Partie pratique**

**Classe Terminale de la voie générale**

---

**Sujet n°15**

---

**DUREE DE L'ÉPREUVE : 1 heure**

**Le sujet comporte 2 pages numérotées de 1 / 2 à 2 / 2  
Dès que le sujet vous est remis, assurez-vous qu'il est complet.**

*Le candidat doit traiter les 2 exercices.*

### EXERCICE 1 (4 points)

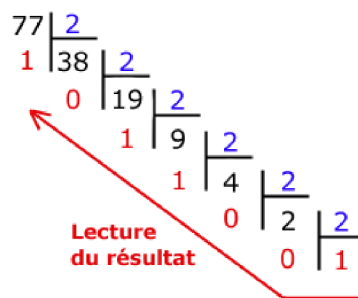
Écrire une fonction python appelée `nb_repetitions` qui prend en paramètres un élément `elt` et une liste `tab` et renvoie le nombre de fois où l'élément apparaît dans la liste.

Exemples :

```
>>> nb_repetitions(5, [2, 5, 3, 5, 6, 9, 5])
3
>>> nb_repetitions('A', ['B', 'A', 'B', 'A', 'R'])
2
>>> nb_repetitions(12, [1, '!', 7, 21, 36, 44])
0
```

### EXERCICE 2 (4 points)

Pour rappel, la conversion d'un nombre entier positif en binaire peut s'effectuer à l'aide des divisions successives comme illustré ici :



Voici une fonction python basée sur la méthode des divisions successives permettant de convertir un nombre entier positif en binaire :

```
def binaire(a):
    bin_a = str(...)
    a = a // 2
    while a ... :
        bin_a = ... (a%2) + ...
        a = ...
    return bin_a
```

Compléter la fonction `binaire`.

Exemples :

```
>>> binaire(0)
'0'
>>> binaire(77)
'1001101'
```

```
def nb_repetitions(elt, tab):  
    somme = 0  
    for x in tab:  
        if x == elt:  
            somme = somme + 1  
    return somme  
  
assert nb_repetitions(5, [2, 5, 3, 5, 6, 9, 5]) == 3  
assert nb_repetitions('A', ['B', 'A', 'B', 'A', 'R']) == 2  
assert nb_repetitions(12, [1, '!', ' ', 7, 21, 36, 44]) == 0
```

```
def binaire(a):  
    bin_a = str(a % 2)  
    a = a // 2  
    while a != 0 :  
        bin_a = str(a % 2) + bin_a  
        a = a // 2  
    return bin_a  
  
assert binaire(0) == '0'  
assert binaire(77) == '1001101'
```