

# BACCALAURÉAT

SESSION 2025

---

Épreuve de l'enseignement de spécialité

## NUMÉRIQUE et SCIENCES INFORMATIQUES

Partie pratique

Classe Terminale de la voie générale

---

Sujet n°27

---

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)

Écrire une fonction `verifie` qui prend en paramètre un tableau de valeurs numériques et qui renvoie `True` si ce tableau est trié dans l'ordre croissant, `False` sinon.

Un tableau vide est considéré comme trié.

Exemples :

```
>>> verifie([0, 5, 8, 8, 9])
True
>>> verifie([8, 12, 4])
False
>>> verifie([-1, 4])
True
>>> verifie([])
True
>>> verifie([5])
True
```

## EXERCICE 2 (10 points)

On considère dans cet exercice l'élection d'un vainqueur à l'issue d'un vote. Les résultats du vote sont stockés dans un tableau : chaque vote exprimé est le nom d'un ou d'une candidate.

Par exemple, les résultats pourraient correspondre au tableau :

```
urne = ['A', 'A', 'A', 'B', 'C', 'B', 'C', 'B', 'C', 'B']
```

indiquant que 3 candidats ont obtenu au moins un vote chacun : A, B et C.

On cherche à déterminer le ou les candidats ayant obtenu le plus de suffrages. Pour cela, on propose d'écrire deux fonctions :

- la fonction `depouille` doit permettre de compter le nombre de votes exprimés pour chacune des issues. Elle prend en paramètre un tableau et renvoie le résultat dans un dictionnaire dont les clés sont les noms des issues et les valeurs le nombre de votes en leur faveur ;
- la fonction `vainqueurs` doit désigner le nom du ou des gagnants. Elle prend en paramètre un dictionnaire **non vide** dont la structure est celle du dictionnaire renvoyé par la fonction `depouille` et renvoie un tableau. Ce tableau peut donc contenir plusieurs éléments s'il y a des artistes ex-aequo.

Compléter les fonctions `depouille` et `vainqueurs` ci-après pour qu'elles renvoient les résultats attendus.

```
def depouille(urne):  
    '''prend en paramètre une liste de suffrages et renvoie un  
    dictionnaire avec le nombre de voix pour chaque candidat'''  
    resultat = ...  
    for bulletin in urne:  
        if ...:  
            resultat[bulletin] = resultat[bulletin] + 1  
        else:  
            ...  
    return resultat  
  
def vainqueurs(election):  
    '''prend en paramètre un dictionnaire non vide avec le nombre  
    ↪ de voix  
    pour chaque candidat et renvoie la liste des vainqueurs'''  
    nmax = 0  
    for candidat in election:  
        if ... > ... :  
            nmax = ...  
    liste_finale = [ nom for nom in election if ... ]  
    return ...
```

Exemples d'utilisation :

```
>>> depouille([ 'A', 'B', 'A' ])
{'A': 2, 'B': 1}
>>> depouille([])
{}
>>> election = depouille(['A', 'A', 'A', 'B', 'C',
                          'B', 'C', 'B', 'C', 'B'])
>>> election
{'A': 3, 'B': 4, 'C': 3}
>>> vainqueurs(election)
['B']
>>> vainqueurs({'A' : 2, 'B' : 2, 'C' : 1})
['A', 'B']
```