

BACCALAURÉAT

SESSION 2024

Épreuve de l'enseignement de spécialité

NUMÉRIQUE et SCIENCES INFORMATIQUES

Partie pratique

Classe Terminale de la voie générale

Sujet n°16

DURÉE 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 (10 points)

Écrire une fonction `écriture_binaire_entier_positif` qui prend en paramètre un entier positif n et renvoie une chaîne de caractères correspondant à l'écriture binaire de n .

On rappelle que :

- l'écriture binaire de 25 est 11001 car $25 = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$;
- $n \% 2$ vaut 0 ou 1 selon que n est pair ou impair ;
- $n // 2$ donne le quotient de la division euclidienne de n par 2.

Il est interdit dans cet exercice d'utiliser la fonction `bin` de Python.

Exemples :

```
>>> 5 % 2
1
>>> 5 // 2
2
>>> écriture_binaire_entier_positif(0)
'0'
>>> écriture_binaire_entier_positif(2)
'10'
>>> écriture_binaire_entier_positif(105)
'1101001'
```

EXERCICE 2 (10 points)

La fonction `tri_bulles` prend en paramètre un tableau `tab` d'entiers (type `list`) et le modifie pour le trier par ordre croissant.

Le tri à bulles est un tri en place qui commence par placer le plus grand élément en dernière position en parcourant le tableau de gauche à droite et en échangeant au passage les éléments voisins mal ordonnés (si la valeur de l'élément d'indice i a une valeur strictement supérieure à celle de l'indice $i + 1$, ils sont échangés). Le tri place ensuite en avant-dernière position le plus grand élément du tableau privé de son dernier élément en procédant encore à des échanges d'éléments voisins. Ce principe est répété jusqu'à placer le minimum en première position.

Exemple : pour trier le tableau `[7, 9, 4, 3]` :

- première étape : 7 et 9 ne sont pas échangés, puis 9 et 4 sont échangés, puis 9 et 3 sont échangés, le tableau est alors `[7, 4, 3, 9]`
- deuxième étape : 7 et 4 sont échangés, puis 7 et 3 sont échangés, le tableau est alors `[4, 3, 7, 9]`
- troisième étape : 4 et 3 sont échangés, le tableau est alors `[3, 4, 7, 9]`

Compléter le code Python ci-dessous qui implémente la fonction `tri_bulles`.

```
def echange(tab, i, j):
    '''Echange les éléments d'indice i et j dans le tableau tab.'''
    temp = ...
    tab[i] = ...
    tab[j] = ...

def tri_bulles(tab):
    '''Trie le tableau tab dans l'ordre croissant
    par la méthode du tri à bulles.'''
    n = len(tab)
    for i in range(...):
        for j in range(...):
            if ... > ...:
                echange(tab, j, ...)
```

Exemples :

```
>>> tab = []
>>> tri_bulles(tab)
>>> tab
[]
>>> tab2 = [9, 3, 7, 2, 3, 1, 6]
>>> tri_bulles(tab2)
>>> tab2
[1, 2, 3, 3, 6, 7, 9]
>>> tab3 = [9, 7, 4, 3]
>>> tri_bulles(tab3)
>>> tab3
[3, 4, 7, 9]
```