

Interrogation écrite  
Bilan sur les tableaux

Durée : 20 minutes

Les calculatrices sont autorisées.

*Consignes.*

- Attendre la fin de la distribution avant de consulter le sujet.
- Écrire ses nom et prénom sur chaque page.
- Répondre aux questions directement sur l'énoncé.

**Exercice 1.** Accès à, modification et ajout d'un élément dans un tableau. (3,5 points)

Pour chaque expression Python ci-dessous, indiquer le résultat d'évaluation dans la console à la place des .... Si une erreur d'indice est levée, écrire simplement **IndexError**.

```

>>> tab1 = [10, 20, 30, 40, 50] ✗
>>> len(tab1)
5.
>>> tab1[2]
30
>>> tab1[-1]
50
>>> tab1[len(tab1)]      tab1[5]
Index Error
>>> tab1[len(tab1) - 1]  tab1[4]
50
>>> tab1[3] = 3
>>> tab1
[10, 20, 30, 3, 50]
>>> tab1.append(70)
>>> tab1.append(60)
>>> tab1
[10, 20, 30, 3, 50, 70, 60]

```

**Exercice 2.** Tableaux en compréhension et construction itérative d'un tableau (3 points)

1. Compléter ci-dessous le résultat d'évaluation dans la console à la place des ....

```

>>> tab2 = [3*i for i in range(5)]
>>> tab2
[0, 3, 6, 9, 12]

```

2. Compléter ci-dessous la déclaration du tableau `tab3` afin qu'elle corresponde au résultat d'évaluation qui suit.

```

>>> tab3 = [2**i for i in range(4, 10)]
>>> tab3
[16, 32, 64, 128, 256, 512]

```

3. Compléter le programme ci-dessous qui construit de manière itérative un tableau `tab4` identique au tableau `tab3` de la question précédente.

```

1 tab4 = []
2 for i in range(4, 10):
3     tab4.append(2**i)

```

**Exercice 3.** Fonctions sur les tableaux. (3,5 points)

On donne ci-dessous la définition d'une fonction `mystere`.

```

1 def mystere(tab):
2     res = 0
3     for i in range(len(tab)):
4         res = res + tab[i]
5     return res

```

1. Compléter ci-dessous le résultat d'évaluation dans la console à la place des `...`.

```

>>> tab5 = [5, 15, 20, 12]
>>> mystere(tab5)
52

```

2. Proposer ci-dessous une documentation pour la fonction `mystere`.

Prend en paramètre un tableau (type `list`) contenant uniquement des nombres, renvoie la somme des éléments de ce tableau.

3. On souhaite écrire la définition d'une fonction `pairs` qui prend en paramètre un tableau (type `list`) contenant uniquement des nombres (type `int` ou `float`). Cette fonction renvoie un nouveau tableau contenant les éléments pairs du tableau passé en paramètre. Voici des exemples d'appels évalués dans la console :

```

>>> tab6 = [12, 5, 7, 8, 13]
>>> pairs(tab6)
[12, 8]
>>> tab7 = [3, 7, 1]
>>> pairs(tab7)
[]

```

Proposer ci-dessous une définition de la fonction `pairs`. Ne pas écrire sa documentation. Attention à soigner les indentations en appliquant un décalage de deux carreaux.

```

def pairs(tab):
    res = [] # On initialise le tableau résultat comme liste vide.
    for i in range(len(tab)): # Pour chaque élément du tableau
        if tab[i] % 2 == 0: # si l'élément courant est pair
            res.append(tab[i]) # on l'ajoute au tableau résultat
    return res

```

↑  
les commentaires ne sont pas exigés.