

Tout document personnel autorisé. Votre voisin n'est pas un document.

---

**Nom :** \_\_\_\_\_ **Prénom :** \_\_\_\_\_

**Signature :** \_\_\_\_\_

Cette feuille contient quatre exercices.

**Consignes à lire attentivement :**

- Placez vous dans un répertoire contenant le fichier `bibTableau.py`.
- Avec `idle3` créer un fichier `nomPrenom.py` où vous remplacerez `nomPrenom` par les vôtres. Dans ce fichier vous écrirez les fonctions demandées ci-dessous ainsi que les appels permettant de les tester.
- Les tests font partie du travail qui est évalué, vos instructions de test doivent donc impérativement figurer dans votre fichier python (tester une fonction signifie ici vérifier son bon fonctionnement sur au moins un exemple). Il faut donc appeler la fonction et afficher son résultat.
- Si une fonction que vous avez écrite ne fonctionne pas, mettez-la en commentaire mais ne l'effacez pas (on pourrait tout de même vous accorder des points si l'idée est pertinente).

**En fin de séance :**

1. Envoyer le fichier par e-mail à votre enseignant(e) de TD, dont l'adresse sera écrite au tableau.
2. Avant de quitter la salle :
  - Se présenter à l'enseignant(e) pour vérifier que l'e-mail est arrivé.
  - Remettre à l'enseignant(e) cette feuille signée (elle attestera de votre présence à l'épreuve).
  - Ne sortez pas de la salle sans l'accord de l'enseignant(e).
3. Garder le fichier contenant votre travail **sans le modifier**. Il pourrait vous être redemandé en cas de problème.

**Exercice 1**

Écrire une fonction `maxIndicesPairs(t,n)` qui retourne la plus grande valeur contenue dans une case d'indice pair du tableau `t` contenant `n` éléments. Si `n` est nul, la fonction doit renvoyer `None`.

**Exercice 2**

Écrire une fonction `supprimerMultiples(t,n,k)` qui modifie le contenu de `t` en supprimant toutes occurrences de multiples de `k` et renvoie le nouveau nombre d'éléments de `t`. La complexité de la fonction sera prise en compte dans la notation.

Exemple :

si `t=[10,3,2,7,15,8,0,12,6,13]`, l'appel de la fonction `supprimerMultiples(t,10,5)` renvoie 7 et on aura `t=[3,2,7,8,12,6,13,12,6,13]`.

### Exercice 3

Soit `t` un tableau contenant `n` nombres. Écrire une fonction `insérerDoublon(t,n,x)` qui cherche la position de la première occurrence de la valeur `x` dans `t` et insère une deuxième occurrence de `x` dans la case immédiatement à sa droite.

La fonction doit renvoyer le nouveau nombre d'éléments de `t`. Si `x` n'est pas dans `t`, le tableau ne sera pas modifié.

On suppose que le tableau passé en paramètre a un nombre d'éléments inférieur à sa taille.

Exemple :

si `t=[10,3,2,7,15,8,0,12,6,13, None, None]`, l'appel de la fonction `insérerDoublon(t,10,7)` renvoie 11 et on aura `t=[10,3,2,7,7,15,8,0,12,6,13, None]`.

### Exercice 4

Soit `t` un tableau de `n` valeurs positives ou négatives. Écrire une fonction `elementSomme(t,n)` qui renvoie `True` si `t` possède une valeur correspondant à la somme de toutes les autres valeurs et renvoie `False` sinon.

Exemple :

l'appel `elementSomme([2,-1,6,5],4)` renverra `True` car `t[2]=t[0]+t[1]+t[3]`, mais `elementSomme([7,8],2)` renverra `False`.