

Module 3 - Aide à la Réussite : EP 2 Soutien en Programmation

Travaux Dirigés (7), Licence 1ère Année

Les Tableaux (suite)

Exercice 1 : Ajout et suppression d'un élément dans un tableau

Pour la suite des exercices suivants, et par commodité, vous aurez besoin d'une méthode qui affiche le contenu d'un tableau d'entiers reçu en paramètre.

Soit un tableau **T** d'éléments entiers non triés.

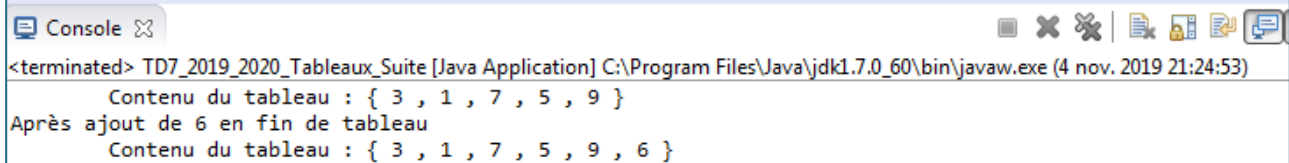
- 1.1. Écrire une méthode **int [] ajoutFin(int [] t, int e)** qui ajoute un élément entier **e** à la fin du tableau **t** et retourne comme résultat le nouveau tableau.

Remarque : vous devrez créer un nouveau tableau.

Note : pour vos tests, vous pouvez déclarer votre tableau **T** ainsi : `int T[] = { 3, 1, 7, 5, 9 }`, par exemple, ce qui vous évitera de l'introduire (remplir) à chaque exécution.

Exemple

```
7      int[] t = { 3, 1, 7, 5, 9 };
8      afficheTab(t);
9      System.out.println("Après ajout de " + 6 + " en fin de tableau");
10     afficheTab(ajoutFin(t, 6));
```



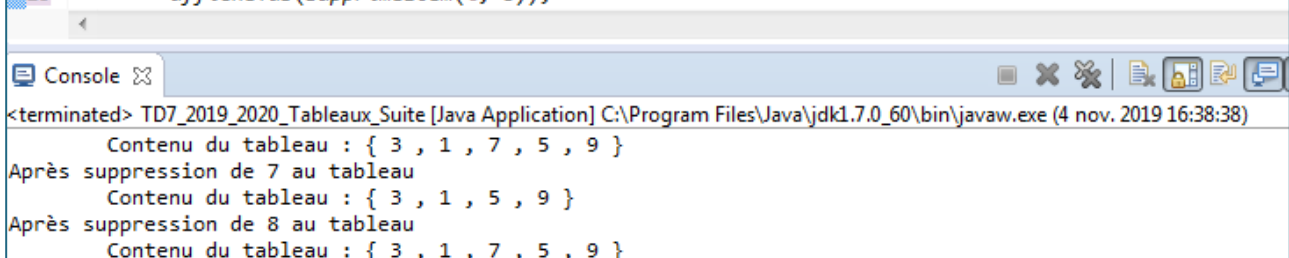
```
<terminated> TD7_2019_2020_Tableaux_Suite [Java Application] C:\Program Files\Java\jdk1.7.0_60\bin\javaw.exe (4 nov. 2019 21:24:53)
Contenu du tableau : { 3 , 1 , 7 , 5 , 9 }
Après ajout de 6 en fin de tableau
Contenu du tableau : { 3 , 1 , 7 , 5 , 9 , 6 }
```

- 1.2. Écrire une méthode **int [] supprimeElem(int [] t, int e)** qui supprime au tableau **t** un élément entier **e**, s'il y est présent, et retourne comme résultat le nouveau tableau. Si l'élément **e** n'est pas présent, la méthode retourne le tableau **t** inchangé.

Remarque : vous devrez créer un nouveau tableau si l'élément est présent.

Exemple

```
20     int[] t = { 3, 1, 7, 5, 9 };
21     afficheTab(t);
22     System.out.println("Après suppression de " + 7 + " au tableau");
23     afficheTab(supprimeElem(t, 7));
24     System.out.println("Après suppression de " + 8 + " au tableau");
25     afficheTab(supprimeElem(t, 8));
```



```
<terminated> TD7_2019_2020_Tableaux_Suite [Java Application] C:\Program Files\Java\jdk1.7.0_60\bin\javaw.exe (4 nov. 2019 16:38:38)
Contenu du tableau : { 3 , 1 , 7 , 5 , 9 }
Après suppression de 7 au tableau
Contenu du tableau : { 3 , 1 , 5 , 9 }
Après suppression de 8 au tableau
Contenu du tableau : { 3 , 1 , 7 , 5 , 9 }
```

On suppose maintenant que le tableau d'entiers **T** est trié ($T[0] \leq T[1] \leq \dots \leq T[n-1]$, où **n** est la taille du tableau **T**).

- 1.3.** Écrire une méthode `int [] ajoutTrie(int [] t, int e)` qui ajoute un élément entier **e** au tableau **t** de façon que ce dernier reste trié.

Exemple

```

27     int[] t = { 1, 3, 5, 7, 9 };
28     afficheTab(t);
29     System.out.println("Après ajout de " + 6 + " au tableau");
30     afficheTab(ajoutTrie(t, 6));
  
```

Console [Java Application] C:\Program Files\Java\jdk1.7.0_60\bin\javaw.exe (4 nov. 2019 20:51:26)
 Contenu du tableau : { 1 , 3 , 5 , 7 , 9 }
 Après ajout de 6 au tableau
 Contenu du tableau : { 1 , 3 , 5 , 6 , 7 , 9 }

- 1.4.** Écrire une méthode `int [] supprimeElemTrie(int [] t, int e)` qui supprime au tableau **t** un élément entier **e**, s'il y est présent, et retourne comme résultat le nouveau tableau. Si l'élément **e** n'est pas présent, la méthode retourne le tableau **t** inchangé.

Exemple

```

32     int[] t = { 1, 3, 5, 7, 9 };
33     afficheTab(t);
34     System.out.println("Après suppression de " + 7 + " au tableau trié");
35     afficheTab(supprimeElem(t, 7));
36     System.out.println("Après suppression de " + 8 + " au tableau trié");
37     afficheTab(supprimeElem(t, 8));
  
```

Console [Java Application] C:\Program Files\Java\jdk1.7.0_60\bin\javaw.exe (4 nov. 2019 21:14:29)
 Contenu du tableau : { 1 , 3 , 5 , 7 , 9 }
 Après suppression de 7 au tableau trié
 Contenu du tableau : { 1 , 3 , 5 , 9 }
 Après suppression de 8 au tableau trié
 Contenu du tableau : { 1 , 3 , 5 , 7 , 9 }

- 1.5.** Estimez-vous que les algorithmes des points **1.1.** et **1.3.** sont équivalents en termes de performances ? Expliquez.
- 1.6.** Estimez-vous que les algorithmes des points **1.2.** et **1.4.** sont équivalents en termes de performances ? Expliquez.