

EXERCICE 18A – TABLEAUX 1D – PARTIE 2

Préparé par Benjamin Lemelin et Pierre Poulin le 22 février 2023

1 Travail à effectuer

Ouvrez le projet fourni avec cet énoncé. Il devrait déjà contenir un fichier **Library.cs**. Il contient aussi des tests unitaires pour chaque fonction à créer. Ces tests sont en commentaires actuellement. Il ne vous restera qu'à les décommenter lorsque nécessaire et à les compléter en fonction des consignes ci-dessous.

1.1 Ajouter à la fin d'un tableau

Créez une fonction nommée **Append** permettant d'insérer un entier à la fin d'un tableau. Pour ce faire, elle crée un nouveau tableau incluant l'entier à insérer et le retourne. Elle nécessite deux paramètres : le tableau dans lequel insérer l'entier et l'entier à insérer.

Par exemple, si nous désirons insérer le nombre 33 dans le tableau suivant :

0	1	2	3	4	5
25	22	56	78	99	45

La fonction retournera ce nouveau tableau :

0	1	2	3	4	5	6
25	22	56	78	99	45	33

Exécutez le test fourni et ajoutez un test pour tester la situation où l'on ajoute dans un tableau vide. Portez attention au nommage de votre test.

1.2 Différence entre deux tableaux

Créez une fonction nommée **Difference** retournant la différence entre deux tableaux. Autrement dit, elle indique quels éléments dans un tableau A ne sont pas dans un tableau B. Par exemple, supposons les deux tableaux suivants :

A						B		
0	1	2	3	4	5	0	1	2
25	22	56	78	33	99	99	56	78

La fonction retournera ce nouveau tableau :

0	1	2
25	22	33

Voici quelques indices pour vous aider à réaliser cette fonction :

- Commencez avec un tableau vide et utilisez votre fonction **Append** pour ajouter un nouvel élément dans votre tableau.
- Pas besoin d'avoir un algorithme très efficace pour cet exercice. Un algorithme de force brute fera le travail. La performance n'est pas un enjeu pour cette question.
- Une fonction **Contains** permettant de déterminer si un élément est dans un tableau risque de vous être fort utile. Vous êtes invités à la créer et à l'utiliser.

Vous disposez déjà de plusieurs tests unitaires pour valider le fonctionnement de votre fonction. En vous inspirant de ceux présents, ajoutez des tests pour les situations suivantes :

- Différence entre deux tableaux possédant les mêmes éléments mais dans un ordre différent.
- Différence entre deux tableaux dont le second possède plus d'éléments que le premier.
[Astuce/indice : il peut y avoir plus d'un cas possible ici]
- Différence de deux tableaux vides

1.3 Inverser un tableau

Créez une fonction nommée **Reverse** qui inverse l'ordre des éléments d'un tableau. Pour être plus exact, elle retourne un **nouveau** tableau contenant les éléments du tableau reçu en paramètre dans l'ordre inverse.

Par exemple, en utilisant le tableau suivant :

0	1	2	3	4	5
25	22	56	78	99	45

La fonction retournera ce nouveau tableau :

0	1	2	3	4	5
45	99	78	56	22	25

Vous disposez d'un test unitaire pour cette fonction. En vous inspirant de ceux présents, ajoutez les tests pour les deux situations suivantes :

- Inversion d'un tableau ne contenant qu'un seul élément
- Inversion d'un tableau vide

2 Modalités de remise

Remettez votre projet Visual Studio sur LÉA, dans la section travaux, à l'intérieur d'une archive *Zip*. Supprimez tous les dossiers temporaires, à savoir les dossiers **.vs**, **TestResults**, **bin** et **obj**.