

EXERCICE 03 – SÉQUENCES

Préparé par Benjamin Lemelin et Pierre Poulin le 5 janvier 2023

1 Travail à effectuer

Ouvrez le projet fourni avec cet énoncé. Il devrait déjà contenir un fichier **Program.cs**.

1.1 Périmètre d'un cercle

Localisez la fonction **CirclePerimeter**. Dans cette fonction, écrivez un algorithme qui calcule le périmètre d'un cercle à partir de son rayon.

Une fois que vous l'aurez fait, vérifiez que votre code fonctionne avec ces tests.

#	Variables	
	Entrées	Sortie
1	radius = 0.0f	circlePerimeter= 0.0
2	radius = 10.0f	circlePerimeter= 62.8318...
3	radius = 5.5f	circlePerimeter= 34.5575...

Vérifiez la valeur en utilisant le débogueur. Certifiez dans les commentaires appropriés que vous avez exécutés ces tests en y inscrivant vos initiales. Indiquez aussi dans les commentaires si chaque cas de test passe.

1.2 Moyenne de trois nombres

Localisez la fonction **NumbersAverage**. Dans cette fonction, écrivez un algorithme qui calcule la moyenne de trois (3) nombres entiers.

Une fois que vous l'aurez fait, vérifiez que votre code fonctionne avec ces tests. Procédez de la même façon qu'au numéro précédent.

#	Variables	
	Entrées	Sortie
1	number1 = 75 number2 = 88 number3 = 53	average= 72
2	number1 = 71 number2 = 66 number3 = 0	average= 45
3	number1 = 0 number2 = 0 number3 = 0	average= 0

1.3 Calcul d'un rabais

Localisez la fonction **ApplyRebate**. Dans cette fonction, écrivez un algorithme qui calcule le nouveau prix d'un produit après un rabais. Le pourcentage de rabais à appliquer se trouve dans la constante **REBATE_RATE** que vous devez utiliser.

Comme toujours, vérifiez que votre code fonctionne avec ces tests.

#	Variables	
	Entrées	Sortie
1	price = 75.0f	amountWithRebate= 67.5
2	price = 0.0f	amountWithRebate= 0.0
3	price = 10.0f	amountWithRebate= 9.0

1.4 Permutations

Localisez la fonction **SwapNumbers**. Dans cette fonction, écrivez un algorithme qui échange la valeur de la variable **number1** avec la variable **number2**.

Vérifiez que votre code fonctionne avec ces tests.

#	Variables	
	Entrées	Sortie
1	number1 = 1 number2 = 2	number1 = 2 number2 = 1
2	number1 = 2 number2 = 1	number1 = 1 number2 = 2
3	number1 = 1 number2 = 1	number1 = 1 number2 = 1

1.5 Calcul des taxes

Localisez la fonction **ComputePriceWithTaxes**. Dans cette fonction, écrivez un algorithme qui calcule le coût d'un produit incluant les taxes (soit la TPS et la TVQ). Les taux de ces deux taxes sont en constante dans le haut du fichier.



Attention

Contrairement à la croyance populaire, la TVQ ne s'applique pas sur la TPS. Les deux taxes sont calculées à partir du coût initial du produit ou du service.

Vérifiez que votre code fonctionne avec ces tests.

#	Variables	
	Entrées	Sortie
1	price = 100.0f	amountWithTaxes= 114.975
2	price = 0.0f	amountWithTaxes= 0.0
3	price = 239.50f	amountWithTaxes= 275.365

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**, **bin** et **obj**.