



Algorithme et programmation (420-1MP-HY)

Travail Pratique #1

Préparé par

Martin Lalancette

bureau B-2335

DESCRIPTION**But :**

Utiliser les éléments de base de la programmation algorithmique.

Objectifs :

- Appliquer la méthode de résolution de problèmes
- Élaborer les solutions via pseudo-code
- Élaborer des cas tests manuels
- Appliquer les principes logiques algorithmiques avec une bonne syntaxe en suivant vigoureusement les **normes de programmation**
- Utiliser les variables de type valeur et les opérateurs
- Utiliser les structures alternatives et opérateurs logiques
- Programmer en C# une application de type console (.NET Core) sous Visual Studio

Durée :

6 h

Pondération :

sur 10

Remise :

À la semaine 4.

Contenu général :

- Remettre vos documents d'analyse et votre solution contenant les projets dans un document .ZIP via **LÉA avant la date et heure limite**.

Notes

- Conserver une copie de sécurité. Il est de **votre responsabilité** de conserver une copie de sécurité dans l'éventualité où la lecture des données serait impossible. Cette copie doit **être disponible sur demande**.

Spécifications du travail

Dans ce travail pratique, il y aura trois parties (A, B et C) à faire **seul ou en équipe de deux**. Les critères suivants seront évalués pour l'ensemble des deux parties :

- Le pseudo-code
- La casse des identificateurs (variables, constantes)
- Le côté significatif des identificateurs

- La lisibilité du code, y compris les commentaires explicatifs
- Le projet doit être compilable facilement
- Coder en C# avec MS Visual Studio 2019 dans un projet de type console

Pour l'ensemble des trois parties, vous devez créer une seule solution nommée TP1.sln, et trois projets de type console .NET Core.

Console 1 : Calculer la moyenne des quantités - automobiles vendues – 25%

On désire avoir un programme qui va demander à un vendeur automobile de saisir le nombre de voitures vendues par trimestre (T1, T2, T3, T4). Ce programme devra calculer la moyenne par trimestre et par mois de ces quantités (valeurs entières) et les afficher. Exemple d'exécution :

```
Calculer la moyenne des quantités - automobiles vendues
Entrer la quantité du trimestre #1: 45
Entrer la quantité du trimestre #2: 56
Entrer la quantité du trimestre #3: 10
Entrer la quantité du trimestre #4: 39
La moyenne par mois est: 12.5
La moyenne par trimestre est: 37.5
Appuyer sur une touche pour quitter...
```

Cas de tests :

Résultat		
Trimestres (T1, T2, T3, T4)	Par trimestre	Par mois
101, 56, 20, 123	75.0	25.0
45, 56, 10, 39	37.5	12.5
104, 207, 96, 77	121.0	40.3

Consignes :

1. Ajouter à la solution un projet de type console (.NET Core) nommé **Console1**.
2. Ajouter un fichier texte (ou word) dans ce projet et y rédiger le pseudo-code.
3. Dans le fichier **Program.cs**, coder la logique de votre pseudo-code en ajoutant des commentaires (Description, Auteur, Date, etc.) et en appliquant les normes.
4. Afficher les moyennes avec 1 décimale.
5. Effectuer les cas (minimum 5) de test pour s'assurer de la bonne fonctionnalité.

Console 2 : Monsieur BargainTout – 25%

Monsieur BargainTout aime se promener dans des ventes de garage pour faire l'acquisition des différents objets. Il aime se vanter qu'il réussit à faire baisser les prix. On désire un programme

qui permet de calculer le pourcentage de rabais suite à un achat. Le programme devra demander le **nom de l'objet**, le **prix affiché** et le **prix payé**. Le programme devra afficher dans une seule fenêtre les informations suivantes : le **nom de l'objet**, le **prix affiché**, le **prix payé**, la **différence entre les 2 prix** et le **pourcentage de rabais** obtenu. Exemple d'exécution:

Cas de tests :

Informations	Résultat(s)
"Tondeuse", 110, 85	25.00 \$, 22.73 %
"Plante", 11.50, 10.30	1.25 \$, 10.43 %
" Tableau Van Gogh", 5999, 4700	1299.00 \$, 21.65 %

Consignes :

1. Ajouter à la solution un projet de type console (.NET Core) nommé **Console2**.
2. Ajouter un fichier texte (ou word) dans ce projet et y réécrire le pseudo-code.
3. Dans le fichier **Program.cs**, coder la logique de votre pseudo-code en ajoutant des commentaires (Description, Auteur, Date, etc.) et en appliquant les normes.
4. Afficher les prix et le pourcentage avec 2 décimales.
5. Effectuer les cas (minimum 5) de test pour s'assurer de la bonne fonctionnalité.

Console 3 : Minimum de monnaie à un client – 50%

On désire un programme qui va planter un algorithme, permettant à une caissière de remettre le minimum de monnaie à un client. C'est-à-dire, un utilisateur achète seulement un article à un prix. L'utilisateur donne des billets à la caissière en sachant que le prix de l'article sera toujours inférieur aux billets reçus (Validation à faire). Le résultat du change est alors affiché à l'écran. Ensuite, la caissière remet un nombre minimum de 5\$, 2\$, 1\$, 25 ¢, 10 ¢, 5 ¢. Vous devez donc afficher à l'écran le nombre de chacun des billets et pièces que la caissière doit remettre et le nombre total de pièce et de billet. Au Canada, les pièces de 1 ¢ ont disparu. Il existe donc des règles d'arrondissement. Vous devez les appliquer à ce programme. Voici les règles :



Exemples d'exécution :

```
Calculer la monnaie
Veuillez entrer le montant de l'article: 23.65
Montant remis par le client: 20
Le client n'a pas donné assez d'argent!
Appuyer sur une touche pour quitter...
```

```
Calculer la monnaie
Veuillez entrer le montant de l'article: 23.65
Montant remis par le client: 40
Vous devez donc lui remettre 16.35 $. Soit:
3 X 5 $
0 X 2 $
1 X 1 $
1 X 0.25 $
1 X 0.10 $
0 X 0.05 $
Ceci fait un total de 6 billets et pièces.
Appuyer sur une touche pour quitter...
```

Consignes :

1. Ajouter à la solution un projet de type console (.NET Core) nommé **Console3**.
2. Ajouter un fichier texte (ou word) dans ce projet et y **rédiger le pseudo-code**.
3. Dans le fichier **Program.cs**, coder la logique de votre pseudo-code en ajoutant des commentaires (Description, Auteur, Date, etc.) et en appliquant les normes.
4. Utiliser des **variables de type decimal** afin d'éviter les pertes de décimales.
5. Afficher les prix **avec 2 décimales**.
6. **Rédiger 5 cas de tests manuels selon la méthodologie vue en classe**.

Barème d'évaluation

Console #1 : Pseudo-code	/1.0
Console #1 : Codification selon les normes.....	/0.5
Console #1 : Programmation : saisie de données, calcul des moyennes et l'affichage	/1.0
Console #2 : Pseudo-code	/1.0
Console #2 : Codification selon les normes.....	/0.5
Console #2 : Programmation : saisie de données, calcul du rabais et l'affichage.....	/1.0
Console #3 : Pseudo-code	/1.0
Console #3 : Codification selon les normes.....	/0.5
Console #3 : Programmation : Validation (montant article < montant client)	/0.3
Console #3 : Programmation : Calculs du nombre de chaque pièce/billet.....	/1.5
Console #3 : Programmation : Compter le nombre de pièces/billets total	/0.2
Console #3 : Programmation : Respect de l'affichage.....	/0.5
Console #3 : Rédaction des 5 cas de tests.....	/1.0
Note totale*	/10.0

PDEA #1: Lors d'activités d'évaluation sommative en classe ou hors classe (programmes (incluant les commentaires), documentation, rapport de laboratoire, rapport de stage, examen), une pénalité **maximale de 10 %** peut être retranchée de la note finale de ladite évaluation (le barème étant **de 0,5%/erreur**).

PDEA #4 : Toute évaluation sommative remise après la date d'échéance fixée se voit attribuer la note zéro pour les étudiants **de 2^e à 6^e session**. Afin de faciliter l'accueil et l'intégration des nouveaux étudiants de **1^{re} session**, cette règle s'appliquera de la façon suivante : 30 % de pénalité pour une 1^{re} offense (à condition que la remise soit faite dans les 24 heures suivant la date de remise officielle), 100% de pénalité pour les offenses subséquentes.

* Tout travail plagié en partie ou en totalité se verra attribuer une note totale de 0%.