

Programmation 1 - 420-F06-RO

- **On doit valider toutes les saisies de données en utilisant le module `valider_nombre` donné dans le cadre de ce cours.**
- Ce travail doit se faire en équipe de deux. La validation sera individuelle.
- Date de la remise : **au plus tard minuit le dimanche 18/12/2022.**
- Date de validation sera donnée ultérieurement et se fera soit en **présentiel**, soit sur **Teams**.
- Vous devez utiliser **Python** pour coder votre programme.

CONSIGNES :

Analyse :

Un fichier texte contenant l'analyse et les jeux de tests de l'exécution de votre programme.

L'analyse devrait être comme suit :

1. **Combien** d'entrées vous avez (informations à saisir) :
 - **Nommer-les** brièvement :
 -
 -
2. **Combien** de données vous avez (informations connues):
 - **Nommer-les** brièvement :
 -
 -
3. **Combien** de résultats à calculer et à afficher :
 - **Nommer-les** brièvement :
 -
 -

Implémentation :

Programme Python

Mettre en commentaire :

- ✓ Un en-tête contenant **vosre nom et prénom et de votre partenaire.**
- ✓ Les noms des variables, des constantes et leurs types et description, Toutes les variables ou constantes doivent être déclarées et décrites **à la fin de votre code en commentaire** (l'équivalent du tableau d'environnement).
- ✓ **N'oubliez pas de placer adéquatement** les **`#end if`**, **`#end while`** et le **`#end for`**

Jeux de tests

Un minimum de 3 jeux de tests : seulement les **affichages sur l'écran** à l'exécution de votre programme Python. PAS de capture écran, utiliser copier-coller à la place

REMISE

Remettre **2 FICHIERS** sur Moodle DecClic **Le fichier Python** de votre programme Python et le fichier **Le fichier texte** ou Word contenant l'analyse comme décrite ci-dessus et 3 bons jeux de tests.

ATTENTION :

Toutes les consignes et les explications décrites dans ce document doivent être respectées et prises en compte dans la codification de votre programme Python.

REMARQUE :

Des **périodes** en classe seront consacrées à travailler, donner des explications et corriger vos programmes.

Énoncé

Après une étude exhaustive faite par des institutions gouvernementales spécialisées dans la fiscalité, Le gouvernement du Québec a conclu qu'il a plusieurs millions de dollars en pertes fiscales à chaque année dans le secteur commercial. Pour contrer ce phénomène, le gouvernement a modifié les règles en obligeant toute transaction doit faire l'objet d'une facture pour tous les clients. La facture est produite au moyen d'un module d'enregistrement des ventes (**MEV**) qui est installé sur toutes les caisses.

On souhaite écrire une partie du logiciel MEV qui permet de produire la facture pour les achats des clients. Votre programme doit calculer et afficher le détail de la facture selon ce que le client a acheté comme articles.

Pour effectuer ces traitements, nous disposons en entrée des informations suivantes :

1. Le nom et prénom du client ;
2. Une liste d'articles achetés par ce client, **pour chaque article** acheté, nous avons :
 - a. Le numéro de l'article (**unique**) ;
 - b. La désignation de l'article (**nom article**) ;
 - c. Le prix unitaire de l'article ;
 - d. La quantité achetée de l'article ;

À l'aide de la saisie de ces informations, votre programme doit produire la facture du client en affichant les informations suivantes :

- A). Nom et prénom du client avec la première lettre en majuscule du nom et du prénom ;
- B). Nombre d'articles achetés
- C). Montant total hors taxe de la facture de ce client ;
- D). Montant total de la TPS de la facture de ce client ;
- E). Montant total de la TVQ de la facture de ce client ;
- F). Montant total toutes taxes comprises de la facture de ce client ;
- G). Pour chaque article de la liste, il faut afficher
 - Numéro d'article
 - Prix unitaire de l'article ;
 - Désignation de l'article ;
 - Quantité achetée de l'article ;
 - Montant hors taxe (Prix unitaire * Quantité achetée) ;
 - Montant TPS de l'article ;
 - Montant TVQ de l'article ;
 - Montant toutes taxes comprises de l'article (TTC) ;

Remarque importante : **AUCUN MONTANT NE DOIT ÊTRE STOCKÉ DANS LA LISTE**, tous les montants sont à calculer à base des informations stockées dans la liste qui sont les prix et les quantités des articles achetés.

Votre programme doit utiliser **UNE LISTE DE LISTES** pour stocker les informations de la facture du client. **Chaque élément de cette liste est une sous liste** représentant un article et stockant les informations de cet article (Numéro **unique**, Désignation, Prix unitaire et Quantité achetée).

Lorsque vous construisez votre liste à partir des informations saisies, elle doit être comme celle décrite ci-dessous :

```
lst_articles = [ [5, 'Clavier', 42.0, 29], [7, 'Souris', 56.0, 25],  
                 [4, 'Écran', 616.0, 220], [3, 'Clé USB', 33.3, 29],  
                 [17, 'Disque', 504.5, 220], [2, 'Cable coaxial', 41.9, 50],  
                 ]
```

Votre programme doit, dès son exécution, demander à l'utilisateur : Le nom et le prénom du client ; puis affiche le menu suivant :

[1]. **Afficher la facture détaillée**

[2]. **Ajouter un article à la facture**

[3]. **Supprimer un article de la facture**

[4]. **Modifier la quantité d'un article**

[5]. **Trier la facture par numéro d'article**

[7]. **Afficher les articles les plus achetés (quantité maximale)**

[8]. **Afficher les articles les moins achetés (quantité minimale)**

[6]. **Afficher sommairement la facture**

[9]. **Quitter**

Saisir votre choix parmi [1..9] :

Le menu décrit ci-dessus, doit être réaffiché jusqu'à ce que l'utilisateur fasse le choix de quitter.

L'option 1 - Afficher - calcule et affiche toutes les informations de la facture (décrites dans les points **A à G**). Si la liste est vide, afficher un message indiquant qu'il n'y a aucun article acheté.

L'option 2 - Ajouter, le programme doit demander à l'utilisateur le numéro d'article (entier >0), la désignation de l'article (chaîne de caractères), le prix unitaire (réel > 0) et la quantité achetée de l'article (entier >= 1). **Si le numéro d'article n'existe pas,** votre programme doit ajouter les informations de l'article à la liste des articles achetés par le client.

Si ce **NUMÉRO EXISTE DÉJÀ**, on demande **seulement** la nouvelle quantité que le client veut ajouter à sa commande et on **l'additionne** à l'ancienne quantité de cet article. La quantité doit être strictement positive.

L'option 3 - Supprimer demande à l'utilisateur le numéro de l'article à supprimer. Ensuite, le programme cherche si ce numéro existe parmi la liste des articles achetés par le client. S'il le trouve, il le supprime, sinon, il affiche un message indiquant qu'aucun article acheté n'a ce numéro.

L'option 4 - Modifier demande à l'utilisateur le numéro de l'article. Si ce numéro existe parmi la liste des articles achetés par le client, votre programme demande la nouvelle quantité (>0) et la met à jour sinon, il affiche un message indiquant que l'article est inexistant.

L'option 5 - Trier doit trier la liste des articles **par ordre croissant des numéros des articles**.

L'option 6 - Afficher les articles les plus achetées affiche la quantité maximale achetée peu importe l'article si la liste est non vide et affiche tous les noms des articles dont la quantité commandée est égale à la quantité maximale.

L'option 7 - Afficher les articles les plus achetées affiche la quantité minimale achetée peu importe l'article si la liste est non vide et affiche tous les noms des articles dont la quantité commandée est égale à la quantité minimale.

L'option 8 - Afficher sommairement – affiche la liste des articles achetés par le client avec un simple print(). Si la liste est vide, afficher un message indiquant qu'il n'y a aucun article acheté par le client.

L'option 9 - Quitter quitte le programme.

Tout choix incorrect doit provoquer l'affichage d'un message d'erreur et le réaffichage du menu. Dans le cas où il n'y a aucun article dans la facture, le programme doit le signaler.

Fonctions :

IL FAUT MINIMALEMENT DÉFINIR LES FONCTIONS SUIVANTES :

- Définir une fonction nommée **menu()** qui **ne prend aucun paramètre**, affiche le menu et retourne le **choix de l'utilisateur**. [retourne un str]
- Définir une fonction nommée **chercher_article()** qui prend **deux paramètres**, la liste des articles et le numéro de l'article et retourne l'indice de l'article s'il est dans la liste sinon retourne -1. Cette fonction est appelée au besoin pour coder les options ajouter, modifier et supprimer décrites ci-dessus. [retourne un int]
- Définir une fonction nommée **ajouter_article()** qui prend **deux paramètres**, la liste des articles et le numéro de l'article à ajouter. Avant d'appeler cette fonction, il faut vérifier si le numéro de l'article existe déjà ou non en utilisant la fonction **chercher_article()**. [retourne rien]
- Définir une fonction nommée **afficher_facture ()** qui prend **un paramètre**, la liste des articles et calcule et affiche les informations demandées dans **les points A) à G)**. [retourne rien]
- Définir une fonction nommée **quantite_maximale()** qui prend **un paramètre**, la liste des articles et retourne la **plus grande** quantité achetée. Avant d'appeler cette fonction, il faut s'assurer que la liste des articles n'est pas vide. [retourne un int]
- Définir une fonction nommée **quantite_minimale()** qui prend **un paramètre**, la liste des articles et retourne la **plus petite** quantité achetée . Avant d'appeler cette fonction, il faut s'assurer que la liste des articles n'est pas vide. [retourne un int]

Grille d'évaluation - Devoir 4	/100
Option 1 - Afficher la facture détaillée	20
Option 2 - Ajouter un article à la facture	15
Option 3 - Supprimer un article de la facture	10
Option 4 - Modifier la quantité d'un article	10
Option 5 - Trier la facture par numéro d'article	10
Option 6 - Afficher les articles les plus achetés	10
Option 7 - Afficher les articles les moins achetés	10
Analyse, affichages explicites et exactitude des affichages demandés	10
Jeux de tests	5
Exactitude des traitements demandés et utilisation des fonctions	Jusqu'à -30%
Erreurs syntaxiques dans le programme	Jusqu'à -30%
Qualité du code	Jusqu'à -20%
Ne pas respecter les consignes	Jusqu'à -20%
Validation de la saisie	Jusqu'à -10%
Qualité du français	Jusqu'à -10%