TRAVAIL PRATIQUE SYNTHÈSE

MISE EN SITUATION

Vous devez créer un petit programme *C++* de caisse enregistreuse pour un marché d'alimentation.

DÉROULEMENT DE L'EXÉCUTION

LECTURE DES FICHIERS

Au lancement, le programme lit premièrement le fichier **Produits.txt** qui contient la liste des produits vendus par le marché d'alimentation. Le format du fichier est décrit ci-dessous. Le fichier n'est lu qu'une seule fois, lors du lancement du programme.

Le programme doit valider tous les éléments de chaque ligne. Si une ligne est invalide, elle est ignorée et le programme passe à la suivante, mais affiche un message indiquant l'erreur et le numéro de la ligne du fichier. Le programme fait une pause après l'affichage de tous les messages d'erreur, avant de passer au menu principal, mais seulement si des erreurs se sont produites.

Si aucun produit n'est lu, un message d'erreur l'indique et le programme se termine.

Si au moins un produit a été lu, le programme lit alors le fichier **Inventaire.txt**. Le format du fichier est décrit ci-dessous. Ce fichier contient la quantité de chaque produit détenu dans l'inventaire du magasin. Le programme valide chaque ligne, mais ignore silencieusement les lignes en erreurs.

Ce n'est pas une erreur de ne pas pouvoir lire ce fichier. Dans ce cas, tous les produits ont une quantité d'inventaire de zéro.

MENU PRINCIPAL

Une fois les fichiers lus, le programme affiche le menu principal suivant.

Notez : le titre du menu doit être adapté pour y mettre votre nom.

```
---------------------------------
  Marché d'alimentation J.Beaudet
---------------------------------

  L) Lister produits
  C) Charger circulaire
  R) Recevoir marchandise
  A) Effectuer achats
  Q) Quitter

Choix:
```

Le programme attend alors le choix de l'utilisateur. Un message d'erreur indique un choix non valide, et le menu est réaffiché. L'écran doit être effacé avant chaque affichage du menu.

OPTION L

Cette option permet d'afficher la liste des produits vendus par le magasin. Elle affiche premièrement un menu permettant de restreindre ou non l'affichage.

```
---------------------------------
  Afficher la liste des produits
---------------------------------

Sélectionnez les produits:
  *) Tous les produits
  C) Circulaire
  Catégorie
    1) Farine et grains
    2) Prêt-à-manger
    3) Viandes et substituts

Choix:
```

Le menu ci-dessus n'est qu'un exemple. Les deux premières options sont toujours présentes, mais la liste de catégories s'adapte aux produits, tels que lus du fichier de produits.

- **`*`**: Permet d'afficher tous les produits vendus par le magasin
- **`C`** : Permet d'afficher uniquement les produits présents sur la circulaire courante, c'est-à-dire ceux pour lesquels une réduction de prix est en cours.
- **`1`**, **`2`**, … : Permet d'afficher uniquement les produits de la catégorie sélectionnée.

Pour chaque produit, on affiche : son code, sa description, son prix, sont rabais s'il y en a un, et sa quantité en inventaire. Si la quantité est zéro, la mention « **`Rupture de stock`** » est affichée au lieu de la quantité. Pour les produits vendus au poids, la quantité est en kilogramme avec **`3`** chiffres après le point, suivi de la mention « kg ».

Au maximum **`10`** produits sont affichés à la fois. Après avoir affiché **`10`** produits, le programme attend une réponse de l'utilisateur. Si l'utilisateur entre « **`R`** », le programme retourne au menu. Sinon, le programme affiche les produits suivants.

Si aucun produit ne correspond à la sélection, le programme affiche simplement : « **`Aucun produit`** ».

OPTION C

Cette option permet de charger une circulaire, c'est-à-dire un fichier contenant une liste de rabais. Le format du fichier est décrit ci-dessous.

Le programme demande premièrement le nom du fichier contenant la liste de rabais, puis va le lire. Si le fichier donné est introuvable, un message d'erreur est affiché et le programme retourne au menu.

Si le fichier est trouvé, tous les rabais actuellement en cours sont annulés. Le programme lit alors le fichier et applique chaque rabais. Le programme valide chaque ligne, mais ignore silencieusement les lignes en erreurs.

Lorsque la lecture du fichier est terminée, le programme affiche le nombre de rabais valides appliqués.

OPTION R

Cette option permet d'augmenter l'inventaire d'un produit.

Le programme demande le code du produit et la quantité à ajouter, puis affiche la nouvelle quantité en inventaire, c'est-à-dire la quantité donnée additionnée à l'ancienne quantité.

Le fichier d'inventaire est alors écrit pour refléter la modification.

OPTION A

Cette option permet d'effectuer des achats et de créer une facture. Initialement, le programme affiche les choix suivants :

```
------------------------
  Effectuer des achats
------------------------

Sous-total: 0.00 $

Indiquez le code du produit, ou
 - 'L' pour lister les produits
 - 'A' pour annuler la facture

>
```

- L'option « **L** » permet d'afficher la liste des produits, comme l'option « **L** » du menu principal.
- L'option « **A** » permet d'annuler la facture et de retourner au menu principal.
  Tant que la facture est vide, l'annulation est automatique. Par contre, à partir du moment où au moins un produit a été ajouté à la facture, une confirmation est requise pour effectuer l'annulation.
- Un code de produit valide permet d'ajouter ce produit à la facture.
- Toute autre chose cause un message d'erreur (« **Choix invalide** » ou « **Produit inconnu** ») et les choix sont réaffichés.

Lorsqu'un code de produit valide est donné, les détails du produit sont affichés : description, quantité, unité, prix, et rabais. Pour les produits mesurés en grammes ou millilitres, le prix pour **100g** ou **100ml** est aussi affiché.

Si le produit est en rupture de stock, un message informe l'utilisateur, et le programme retourne au menu des achats. Il faut prendre en compte les produits déjà ajoutés à la facture. Sinon, il demande la quantité de produit voulu. Ici, il y a deux possibilités :

- Pour un produit vendu au poids, le programme demande le poids voulu et lit une valeur réelle.
- Pour les autres produits, le programme demande la quantité voulue et lit une valeur entière.

Le programme lit la valeur donnée par l'utilisateur.

Si l'inventaire n'est pas suffisant pour la quantité demandée, le programme affiche un message et retourne aux options d'achat. Sinon, il affiche le prix pour cette quantité de ce produit. La mention « **(T)** » est ajoutée si le produit est taxable.

Après une pause, les choix sont réaffichés, avec le sous-total mis à jour.

Lorsqu'au moins un produit a été ajouté à la facture, deux options supplémentaires sont offertes.

```
------------------------
  Effectuer des achats
------------------------

Sous-total: 17.79 $

Indiquez le code du produit, ou
 - 'R' pour retirer le dernier produit ajouté
 - 'P' pour payer
 - 'L' pour lister les produits
 - 'A' pour annuler la facture

>
```

- L'option « `R` » permet de retirer le dernier produit ajouté à la facture. Une confirmation est requise pour effectuer le retrait. Lorsqu'un produit est retiré de la facture, il devient possible d'en retirer un autre, et ce jusqu'à ce que la facture soit vide.
- L'option « `P` » permet de finaliser la transaction et de payer la facture.

OPTION P

Lorsque l'option « `P` » est sélectionnée, le programme affiche alors un résumé de la transaction en listant tous les produits ajoutés à la facture. Par exemple :

```
---------------------------
  Finaliser la transaction
---------------------------

Description                      Quantité        Prix            Total
=============================================================================
Banane                          2.150 kg     1.96 $           4.21 $
Lait sans lactose 3,25 %               3     6.69 $          20.07 $
Eau minérale naturelle gazéifiée       1     3.29 $           3.29 $ (T)
=============================================================================
                                          Sous-total:          27.57 $
                                                TPS:           0.16 $
                                                TVQ:           0.33 $
                                               Total:          28.06 $
                                    Montant à payer:           28.05 $
Argent payé:
```

Les produits sont affichés dans l'ordre qu'ils ont été ajoutés à la facture. Il est possible pour un produit d'être affiché plusieurs fois.

Si,et seulement si au moins un produit taxable est ajouté à la facture, le programme affiche la TPS (`5%` du sous-total) et TVQ (`9,975%` du sous-total). Le total est alors la somme du sous-total, de la TPS et de la TVQ. Sinon, le total est égal au sous-total.

Puisque la première version du système ne supporte que les paiements en argent comptant, le « **Montant à payer** » est le montant total arrondi à un multiple de `5¢` selon les règles d'arrondissement (Élimination de la pièce d'un cent).

Le programme attend alors que l'utilisateur indique le montant d'argent donné par le client pour payer l'achat, et valide ce montant.

Si le montant est zéro ou négatif, ou n'est pas un multiple de `5$`, un message indique qu'il est invalide, et retourne au résumé de la transaction pour demander un autre montant.

Si le montant est plus petit que le total, un message indique qu'il est insuffisant, et retourne au résumé de la transaction pour demander un autre montant.

Si le montant est valide et suffisant, le programme affiche la monnaie à rendre et décompose cette valeur pour les différents billets et pièces possibles : `20$`, `10$`, `5$`, `2$`, `1$`. `0.25$`, `0.10$` et `0.05$`.

- Le nombre minimal de billets et pièces doit être retourné. Ainsi, si le montant à rendre est de `20$`, il faut rendre `1` billet de `20$` et non pas `2` billets de `10$` ou `4` billets de `5$`.
- De plus, votre programme doit afficher seulement les billets et pièces à rendre, il ne doit pas indiquer les billets et pièces si leur nombre est zéro.

Une fois la transaction terminée, le fichier d'inventaire est écrit pour refléter la modification et le programme retourne au menu principal.

Option Q

Cette option termine le programme.

Validation des données de l'utilisateur

Le programme doit résister à toute entrée faite par l'utilisateur.

Pour tout choix disponible, le programme accepte les lettres minuscules et majuscules.

Les espaces au début et à la fin d'une entrée sont toujours ignorées.

Dans les menus, un et un seul caractère est accepté. Si aucun ou plus d'un caractère n'est donné, le choix est invalide.

Lors d'une confirmation, la lettre '`o`' permet de répondre « oui ». N'importe quoi d'autre répond « non ».

Pour un nombre entier ou réel, aucun autre caractère ne doit être présent dans la réponse après le nombre.

Format du fichier « `produits.txt` »

Les lignes débutant par '`#`' (il peut y avoir des espaces avant, au début de la ligne) sont des commentaires qui sont ignorés par le programme. Les lignes vides ou ne contenant que des espaces sont aussi ignorées par le programme.

Au début du fichier fourni avec cet énoncé, un commentaire explique le format de chaque ligne.

Ce fichier n'est lu qu'une seule fois, au lancement du programme, et n'est jamais modifié par le programme.

Format du fichier « `inventaire.txt` »

Chaque ligne du fichier contient deux éléments, séparés par un symbole deux-points (`:`) : le code du produit et la quantité en inventaire (nombre entier). Dans le cas des produits vendus au poids, la quantité représente des grammes. La quantité doit être plus grande que zéro.

Un code de produit inconnu (non présent dans le fichier des produits) est considéré invalide et est ignoré. Si deux ou plusieurs quantités sont définies pour le même produit, la dernière lue remplace les précédentes.

Ce fichier n'est lu qu'une seule fois, au lancement du programme. Il est réécrit par le programme après chaque modification de l'inventaire (réception de marchandise, ou paiement de facture). Le fichier n'est pas modifié pendant la phase des achats puisque ceux-ci peuvent être annulés.

Lors de l'écriture, seules les quantités plus grandes que zéro sont ajoutées au fichier. Un produit ne doit pas apparaitre plus d'une fois dans le fichier.

Format d'un fichier de circulaire

Chaque ligne du fichier contient deux éléments, séparés par une virgule (`,`) : le code du produit et le pourcentage de rabais (nombre entier). Ce nombre doit être plus grand que zéro et plus petit ou égal à cent.

Un code de produit inconnu (non présent dans le fichier des produits) est considéré invalide et est ignoré. Si deux ou plusieurs rabais sont définis pour le même produit, le dernier lu remplace les précédents.

Un fichier de circulaire n'est lu que lorsque l'option de charger une circulaire est sélectionnée par l'utilisateur.

Exemple d'exécution

Joint à cet énoncé, vous trouverez le fichier exécutable **ExempleTPSynthese.exe**. Utilisez-le comme référence. Votre programme doit se comporter **exactement** comme celui-ci. Portez une attention particulière à l'information affichée à chaque étape, et aux formats d'affichage, entre autres l'alignement des éléments en colonnes. L'affichage de votre programme doit être identique à cette référence.

Exigences

- Définir des constantes pour les valeurs utilisées à plusieurs reprises dans le programme.
- Utiliser une structure pour conserver toute l'information d'un produit.
- Utiliser un vecteur pour conserver tous les produits définis.

Fichiers

Votre programme doit être divisé en différents fichiers de source et d'en-tête, de façon logique, selon les catégories de fonctionnalités.

La fonction `main` doit être seule dans un fichier.

Fonctions

Votre programme doit limiter autant que possible la duplication de code et opter pour la définition de fonctions qui permettront la réutilisation du code. Il devra aussi découper en plus petites fonctions les tâches longues et complexes.

Commentaires et Normes de programmation

Votre programme doit respecter les normes de programmation et les normes de commentaires décrites dans le document **Normes de programmation.pdf**, disponible dans le dossier partagé du cours.

Pondération

Ce travail compte pour **15** % de la note finale

- Fonctionnement correct du programme 40 %
  - Un programme qui ne compile pas entraine automatiquement **0** pour ce critère
- Choix approprié des types de données élémentaires 10 %
- Découpage efficace du code informatique 15 %
- Choix et organisation logiques des instructions 10 %
- Notation claire et pertinente de commentaires en nombre approprié dans le code informatique 20 %
- Respect systématique des normes de programmation 5 %

Pénalités

Instruction inconnue : Pénalité de **10** % par notion ou instruction non vue en classe.

Français : Jusqu'à 10 % de pénalité

Retard : Pénalité de 10 % par jour de retard (incluant les fins de semaine)

Plagiat : Ceci est un travail individuel. Application de la Politique institutionnelle pour *toutes les personnes* impliquées.

Consignes

- Sur le site *GitHub*, créez un nouveau dépôt **privé** nommé « **`TPSyntheseProgStructuree`** ».
  - Dans l'onglet « **Settings** », dans la catégorie « **Collaborators** », cliquez le bouton vert « **Add people** ».
  - Dans la fenêtre qui s'ouvre, entrez « **`jbeaudet-cstj`** », puis cliquez le bouton « **Add jbeaudet-cstj to this repository** ».
- Clonez ce nouveau dépôt sur votre ordinateur.
- Dans *Visual Studio*, créez un nouveau projet vide.
  - Nommez-le projet « **`ProjetTPSynthese`** ».
  - Choisissez votre dépôt « **`TPSyntheseProgStructuree`** » comme emplacement.
  - Nommez-la solution « **`SolutionTPSynthese`** ».
  - Ajoutez et créez les fichiers requis.

Ne laissez pas votre travail dans les laboratoires

Les disques des ordinateurs des laboratoires sont accessibles à tous. Si vous y laissez votre travail et que quelqu'un d'autre y accède, vous pourriez vous retrouver dans une situation de plagiat.

Assurez-vous de sauvegarder votre travail sur *GitHub*, puis supprimez tout du disque local.

Remise

Retrouvez (dans l'historique de *GitHub* ou à l'aide de la commande « **`git log`** ») l'identifiant du commit correspondant à votre remise. Copiez-le.

Ouvrez l'application « **`Bloc-Notes`** » (*notepad*) et collez l'identifiant copié. Sauvegardez le fichier.

Vous devez remettre le fichier contenant l'identifiant dans LÉA, dans la rubrique Travaux, pour le TP synthèse.

Vous devez remettre votre travail avant **`08:00`** le **`mercredi 11 décembre 2024`**.

Aucun retard ne sera accepté