# 420-135-SH

# Introduction à la programmation

# Travail de session

# Automne 2019

**Consignes**

* Vous devez faire le travail en équipe de 2 ou 3 étudiants.
* Vous pouvez utiliser le langage C ou C++, à votre choix.
* Le travail compte pour 40% de la note finale.
* Le barème de correction suit l’énoncé.
* Vous avez six périodes de cours pour faire le travail. La date de remise est le 12 décembre à 8h30 pour le groupe 4119 et le 13 décembre à 8h30 pour les groupes 4120 et 4121, sauf si un cours est reporté.
* Vous devez trouver par vous-même comment utiliser la librairie string.h avec les fonctions strcat(), srtrlen(), strcmp() et strcpy().
* Aucune variable globale n’est autorisée. Les constantes globales le sont.

**Énoncé**

Le propriétaire d’un restaurant McDonald’s vous demande de lui faire un programme pour la gestion de ses inventaires, principalement pour calculer les quantités vendues et les quantités perdues à chaque jour.

À chaque soir, un employé fait l’inventaire des quantités de nourriture restantes dans le restaurant. Il inscrit ses observations dans un fichier dont le nom a le format « Inventaires\_AAAA\_MM\_JJ.txt ». Le format du fichier est : <NomProduit;quantité;unité>, c’est-à-dire que l’employé inscrit, un produit par ligne, le nom du produit, suivi du point-virgule, suivi de la quantité, suivi du point-virgule, suivi de l’unité. Par exemple, « Boeuf;542,4;kg » signifie qu’il a trouvé 542,2 kg de bœuf dans le frigo. Parfois, les produits sont par paquets. À ce moment, il ne compte pas toutes les unités. Il compte les paquets non ouverts et les unités déballées. Par exemple, « Pain;360(12) + 5;unité » signifie qu’il a trouvé 360 paquets de 12 pains et 5 pains individuels. Vous avez un exemple de fichier nommé « Inventaires\_2019\_11\_19.txt ».

De la même façon, un fichier dont le nom a le format « Achats\_AAAA\_MM\_JJ.txt » sert à enregistrer les ingrédients reçus dans la journée. Le format du fichier est le même : <NomProduit;quantité;unité>. Par contre, les produits reçus par paquets ne contiennent pas d’unités déballées. Par exemple, « Pain;70(12);unité » signifie qu’on a reçu 70 paquets de 12 pains. Vous avez un exemple de fichier nommé « Achat\_2019\_11\_20.txt ».

Finalement, la caisse enregistreuse produit, à chaque jour, un fichier dont le nom a le format « Ventes\_AAAA\_MM\_JJ.txt ». Le format du fichier est : <NomProduit;montant>, c’est-à-dire que la caisse inscrit, un produit par ligne, le nom du produit, suivi du point-virgule, suivi du **montant en argent** des ventes. Vous avez un exemple avec le fichier « Ventes\_2019\_11\_20.txt ».

Vous devez produire un programme qui demande la date, en format AAAA\_MM\_JJ, puis qui trouve les fichiers nécessaires pour faire vos calculs. Vous devez calculer, pour chaque ingrédient, la quantité utilisée durant la journée, ainsi que la quantité perdue (nourriture échappée, produit refusé par un client, etc.). Pour ce faire, vous devez utiliser la formule suivante :

|  |  |
| --- | --- |
| Inventaire précédent | Inventaire |
| plus achat du jour | Achats |
| moins quantité utilisée du jour | Ventes |
| moins perte du jour | Inventaire - ventes |
| = inventaire final du jour. | Analyse |

Remarquez que vous disposez des deux fichiers d’inventaires (précédent et final) et que vous devez utiliser la formule pour calculer les quantités perdues. Pour vous aider, le propriétaire vous a fourni le fichier « Ingredients.xlsx » qui contient les ingrédients ainsi que les prix de chaque produit vendu. Vous pouvez inscrire ces données dans des constantes dans votre programme.

Votre programme doit produire le rapport suivant, autant à l’écran que dans un fichier dont le nom a le format « Analyse\_AAAA\_MM\_JJ.txt » :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Quantité au début | Quantité achetée | Quantité utilisée | Quantité perdue | Quantité à la fin |
| Ingrédient | xxx | xxx | xxx | xxx | xxx |
| Ingrédient | xxx | xxx | xxx | xxx | xxx |
| Ingrédient | xxx | xxx | xxx | xxx | xxx |
| Ingrédient | xxx | xxx | xxx | xxx | xxx |
| Ingrédient | xxx | xxx | xxx | xxx | xxx |

**Barème**

* Fonctionnement correct.
  + (7 points) Lecture des fichiers.
  + (7 points) Calculs exacts.
  + (7 points) Écriture des rapports.
* (7 points) Division optimale du code en fonction. Pas de répétitions.
* (6 points) Utilisation juste des tableaux.
* (6 points) Commentaires exacts et complets.