

IFT1169 – TRAVAIL PRATIQUE #2 – 08 octobre 2020**« Un peu de magasinage ... »
Mohamed Lokbani**

Équipes : le travail est à faire en monôme (une seule personne).

Remise : une seule remise est à effectuer par voie électronique **le mardi 03 novembre 2020, 23h59 au plus tard**, sans possibilités de prorogation.

Conseils : n'attendez pas le dernier jour avant la remise pour engager votre travail. Vous n'aurez pas le temps nécessaire pour le réaliser.

But : ce TP a pour but de vous faire pratiquer la gestion des options, les exceptions et la surcharge des opérateurs.

Énoncé : ce travail est subdivisé sur plusieurs étapes vous permettant de construire graduellement le programme demandé dans ce travail.

Vous allez définir un magasin de commerce et charger dans son ordinateur central l'ensemble des articles qu'il peut contenir.

Vous ne pouvez pas utiliser ni les classes (ou les fonctions) génériques ni les « STL ». Il vous est possible par contre d'utiliser la classe « string » et ses fonctionnalités. Pour plus de détails sur la manipulation des chaînes de caractères du type « string » ou « char* », examiner les deux exemples « string911.cpp » et « char911.cpp », disponibles sur le site web du cours.

Étape -1- : écrire un programme dans lequel la fonction « main » doit traiter et analyser les options suivantes :

- « -h » : affiche une description sommaire des fonctionnalités du programme et ses options, puis elle quitte le programme.
- « -m » : affiche une fonctionnalité quelconque du système et donne la main au programme pour poursuivre son exécution jusqu'à ce qu'il prenne fin.
- « -r » : cette option prend un argument. À ce stade, on se contente d'afficher l'argument fourni. Le programme poursuit par la suite son exécution jusqu'à ce qu'il prenne fin.

À noter : si les 3 options sont présentes, l'option « -h » a la priorité du traitement, vu sa fonction. Les autres options ne seront pas analysées puisque, comme cela a été indiqué, le programme prend fin juste après le traitement de l'option « -h ».

Étape -2- : nous allons reprendre le programme de l'étape -1- et lui ajouter le traitement des redirections telles que décrites dans le document « Redirection.pdf ».

Le programme doit-être capable de lire les données du fichier « entree.txt » et de les afficher en sortie. Cette sortie est par la suite redirigée vers le fichier « sortie.txt ».

tp2.exe < entree.txt > sortie.txt

Nous voulons que ce traitement prenne en considération la présence ou non des options sur la ligne de commandes, tout en respectant les consignes d'utilisation de l'étape -1-.

Ainsi donc :

-1- « tp2.exe -h -m < entree.txt > sortie.txt »

L'option « -h » a la priorité du traitement et donc le programme va prendre fin sans traiter les redirections. Dans ce cas, l'affichage sur la console est redirigé vers le fichier « sortie.txt ».

-2- « tp2.exe -m < entree.txt > sortie.txt »

L'option « -m » se contente d'afficher un message court et laisse le programme poursuivre son traitement. Ce traitement consiste à lire le contenu du fichier « entree.txt ».

L'affichage vers la console est redirigé vers le fichier « sortie.txt ». Ce fichier va contenir d'abord le texte affiché suite à l'utilisation de l'option « -m », par la suite le contenu du fichier « entree.txt ».

À ce stade de l'exercice, il en est de même pour l'option « -r ».

Étape -3- : le fichier « entree.txt » contient ce qui suit :

« entree.txt »
:M 08:00 20:00 Canadian Tire Montréal

L'entrée magasin dans le fichier de données est représentée par «:M ». On associe à un magasin **obligatoirement** les informations suivantes : l'heure d'ouverture, l'heure de fermeture, son nom et la localité où il se trouve.

-1- Ajuster l'option « -m » pour afficher les paramètres associés au magasin.

Il arrive parfois qu'il y ait une erreur de saisie de données dans le fichier « entree.txt ». La colonne de gauche du tableau ci-dessous contient toutes les informations nécessaires pour représenter un magasin, ce qui n'est pas le cas de la colonne de droite.

« entree.txt »	
Fichier correct	Fichier incorrect
:M 08:00 20:00 Canadian Tire Montréal	:M 08:00 20:00 Montréal

Le nom du magasin est manquant. Dans ce cas, on lève une exception pour signaler l'élément manquant et on sort du programme.

-2- Ajuster le précédent programme pour valider le contenu du fichier.

Étape -4- : le fichier « entree.txt » va contenir en réalité deux types d'informations.

«:M » pour signifier un magasin dont les éléments ont été décrits dans l'étape -3-. Le fichier ne va contenir qu'un seul magasin et ce dernier n'est pas obligatoirement le premier défini dans le fichier.

«:A » pour signifier ajouter un article dans l'inventaire du magasin. Un article a la représentation suivante :

Nom
Code-barre pour l'identifier
Prix
Quantité disponible

Un article peut avoir une quantité nulle, dans ce cas, il n'est pas en stock.

Le nombre d'articles disponibles dans le magasin n'est pas défini au départ. Le magasin ne peut pas avoir deux articles ayant un code-barre identique. Si le fichier en entrée contient une telle erreur, ou que l'un des éléments de cet article est manquant, une exception est levée pour signaler cette anomalie.

Ajuster le programme de l'étape -3- pour qu'il prenne en considération des modifications mentionnées dans cette 4^e étape.

Étape -5- : nous allons compléter le fichier « entree.txt » avec l'entrée «:E » comme suit :

«:E » pour signifier retirer un article du magasin. L'article n'est plus fabriqué par exemple. L'article à retirer est identifié par son code-barre unique.

:E
123456

Par ailleurs, nous allons traiter le cas de l'option « -r ». Cette option permet de rechercher un article dans le magasin et afficher ses informations. Comme l'option « -r » prend un argument, ce dernier peut-être le nom du produit ou son code-barre.

Si l'article n'a jamais été inventorié dans le magasin, on informe que l'article n'est pas disponible pour le moment. Dans le cas où l'article est répertorié, mais qu'il n'est pas disponible en stock (quantité est nulle), on affiche un message en conséquence.

Un exemple d'utilisation de l'option « -r » :

tp2.exe -r 12345

Codelite : assurez-vous d'avoir activé dans codelite ces options : « Wall », « pedantic » et « c++17 ».

Fichiers à remettre :

« TP2_A20.workspace » : tout le travail doit-être réalisé dans codelite dans l'espace de travail (« workspace ») « TP2_A20 ». Chaque étape de ce travail pratique est matérialisée dans un projet. Ainsi l'espace de travail va contenir au final 5 projets.

« TP2_A20 _Rapport.pdf » : le rapport doit décrire le travail effectué.

« TP2_A20 _feuille2route.pdf » : une auto-évaluation du travail réalisé.

Vous devez regrouper l'ensemble de ces fichiers dans « TP2_A20.zip » et le remettre par l'entremise du système Studium, dans le dépôt « tp02 ».

La correction va se faire sur un poste de la DESI en utilisant le compilateur « g++ » (9.2). Vérifiez que le code que vous avez produit (devant normalement fonctionner correctement chez vous) fonctionne aussi bien sur les ordinateurs de la DESI.

Barème : ce TP2 est noté sur **12 points**. Les points sont répartis comme suit :

Étape -1- : traitement des options (**1 point**), attention à l'ordre demandé.

Étape -2- : traitement des redirections (**1 point**), attention à l'ordre demandé.

Étape -3- : traitement de l'option « -m » (**1 point**), traitement des erreurs (**1 point**)

Étape -4- : structure des classes (**2 points**), ajouter un article (opérateur +) et la gestion des erreurs (**2 points**)

Étape -5- : retirer un article (opérateur -) (**1 point**), traitement de l'option « -r » (**2 points**).

Rapport : **1 point**

La correction se fera en fonction des composants implémentés dans le programme. Par exemple, s'il vous manque l'option « -r » de l'étape -5-, vous n'allez perdre que deux points. Par contre si vous omettez d'introduire l'option « -r » comme cela a été demandé dans l'étape -1-, vous perdez aussi « 0.25 ».

En plus du précédent barème, vous risquez de perdre des points dans les cas suivants

- La non-remise électronique (volontaire ou par erreur) est sanctionnée par la note 0.
- La non-remise du fichier « TP2_A20 _feuille2route.pdf » : -1.
- Un programme qui ne compile pas : 0.
- Un programme qui compile, mais ne réalise pas les choses prévues dans la spécification : 0.
- Les avertissements (warnings) non corrigés : cela dépend de la quantité! À partir de -0.25 et plus.
- Aberration dans le codage : même si tous les chemins mènent à Rome, faites l'effort nécessaire pour éviter de prendre le plus long!

Des questions à propos de ce TP? Une seule adresse : dift1169@iro.umontreal.ca

Pour faciliter le traitement de votre requête, inclure dans le sujet de votre courriel, au moins la chaîne [IFT1169] et une référence au tp02.

Mise à jour

08-10-2020 diffusion de l'énoncé.

Annexe -1-
Fichiers « entree.txt » incorrect

	« entree.txt »
On a omis l'heure de fermeture!	:M 08:00 Canadian Tire Montréal

	« entree.txt »
On a omis l'heure de fermeture et la localité!	:M 08:00 Canadian Tire

Un ou deux exemples suffisent pour valider votre conception de la gestion des erreurs. Rien ne vous empêche de tester d'autres cas de figure pour bétonner votre conception.

Annexe -2-
Fichiers « entree.txt »

« entree.txt »
:M 08:00 22:00 Canadian Tire Montréal :A Poêle 12345 24.99 100 :A Perceuse 14563 131.74 30