

IFT1169 – TRAVAIL PRATIQUE #3 – 2 novembre 2015**« La STL et ses outils »
Mohamed Lokbani**

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

Remise : une seule remise est à effectuer par voie électronique **le vendredi 18 décembre 2015, 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 l'utilisation des différents outils de la STL.

Notes importantes :

-1- Cette remarque, je n'avais vraiment pas besoin de la faire, pensant faire appel à votre bon sens! La composante principale du cours « IFT1169 » est la programmation orientée objet (P.O.O). On attend de vous à réaliser les travaux pratiques en utilisant les différentes techniques de « P.O.O ». Utiliser une programmation procédurale pour réaliser ce travail est un non-sens. La question n'est pas de savoir s'il est possible ou non d'utiliser la programmation procédurale, la question ne devant même pas être soulevée. Vous ne suivez pas un cours de « C », mais un cours de « C++ ». De ce fait, si vous avez réalisé votre TP#2 en utilisant la programmation procédurale, vous n'avez le choix que de le reprendre de A à Z pour le rendre conforme. Ce n'est pas juste de mettre toutes les fonctions dans une classe et que l'on ne parle plus. J'attends de vous un vrai travail de fond sur le volet « P.O.O ».

-2- Vous ne codez pas pour vous, ni pour le correcteur. Vous codez pour quelqu'un qui n'a qu'une idée sommaire de l'utilité de votre programme. De ce fait, votre code doit contenir les commentaires nécessaires pour aider le lecteur à comprendre ce que vous essayez de faire. On ne vous demande pas de nous dire que « cette instruction » est une boucle. On vous demande de nous expliquer son utilité. Il en est de même dans le cas d'une fonction ou d'une méthode, etc.

-3- Un rapport n'est pas là pour expliquer vos états d'âme. Un rapport doit expliquer votre démarche, décortiquer la structure de votre programme, qui fait quoi et pourquoi, comment utiliser votre programme, etc. Dans la section « FAQ », sur le site web du cours, il y a déjà un document qui vous explique plus en détail, le contenu d'un rapport.

Énoncé : vous allez reprendre le travail #2 pour lui apporter les modifications suivantes :

-1- Vous allez remplacer les structures de données utilisées par des conteneurs de la STL les plus appropriés. Vous devez nous expliquer dans le rapport les raisons (objectives) qui vous ont poussé à choisir le conteneur X au lieu du conteneur Y.

-2- Vous allez maximiser l'utilisation des algorithmes de la STL, si ça se justifie.

-3- On n'utilise plus la redirection pour lire le fichier « fichier_entree.txt ». Le nom du fichier est fourni au programme à travers l'option « -in ». Cette option accepte comme argument le nom du fichier contenant les données à analyser : « -in nom_fichier ». Un message d'erreur approprié est affiché si l'option est absente ou s'il y a une erreur de lecture. L'affichage en sortie ne change pas. Les données sont soit affichées à l'écran ou bien elles sont redirigées vers un fichier en utilisant le signe « > ».

-4- On vous demande d'ajouter l'option permettant de trier le résultat du programme. Pour cela vous allez utiliser l'option « -sort ». Cette option peut prendre une valeur positive dans le cas d'un tri ascendant (petit/grand), ou une valeur négative dans le cas d'un tri descendant (grand/petit). Le tri va se faire sur 1 des 9 champs disponibles. On vous rappelle la délimitation des 9 champs :

```
4334123 4 drwx----- 2 cal12345 etud 4096 Dec 31 1969 ./orbit-cal12345
```

1	2	3	4	5	6	7	8	9
4334123	4	drwx-----	2	cal12345	etud	4096	Dec 31 1969	./orbit-cal12345

-sort +1 : tri ascendant du résultat sur les éléments de la 1^e colonne.

-sort -7 : tri descendant du résultat sur les éléments de la 7^e colonne.

Makefile

L'ensemble des commandes de compilation doit figurer dans le fichier « makefile ». Vous devez inclure aussi dans ce fichier les différentes commandes « diff » décrites dans ce travail et nécessaires pour valider votre travail.

Fichiers à remettre : vous devez remettre le fichier « tp3A15.zip ». Par ailleurs, vous devez inclure dans votre remise un rapport décrivant le travail effectué.

Remise : vérifiez que le code que vous avez produit (devant normalement fonctionner correctement chez vous) fonctionne aussi bien sur les ordinateurs de la DESI. La procédure a déjà été décrite dans le TP#0.

Le répertoire de remise sur le serveur est tp03.

Barème : ce TP3 est noté sur **10 points**. Les points sont répartis comme suit :

STL (conteneurs et algorithmes) : **4 points (incluant la présentation et les commentaires)**

Tris : **4 points (incluant la présentation et les commentaires)**

Rapport : **2 points**

Makefile : **-2 points s'il est absent**

O.O.P : **-5 points si elle est absente**

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.
- 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 tp03.

Mise à jour

02-11-2015 diffusion de l'énoncé.