

IFT1169 – TRAVAIL PRATIQUE #4 – 08 avril 2006

La STL à toutes les sauces !

Mohamed Lokbani

Équipes: Le travail peut-être fait en binôme. Vous ne remettez alors qu'un travail par équipe.

Remise : Deux remises à effectuer : électronique et papier le **vendredi 28 avril 2006, 09h00 au plus tard, sans possibilités de retard**. Glisser votre rapport sous la porte de mon bureau au P-A-A, 3^e étage, 3225.

Conseils: N'attendez pas le dernier jour avant la remise pour commencer, vous n'aurez pas le temps nécessaire pour le faire.

Remarques : Les locaux de la DESI fermeront le 20 avril au soir. Les ordinateurs de la salle « 3311 » au 3^e étage du pavillon André-Aisenstadt vous seront accessibles à partir du 21 avril, de manière exceptionnelle et cela jusqu'à la fin de la session.

But : Ce TP est composé de trois exercices. Il a pour but de vous faire connaître les différents conteneurs de la STL et les méthodes associées à ces conteneurs.

Exercice 1 : En utilisant les STL, écrire le programme « tp4exo1.cpp (et le « .h » associé si nécessaire) » qui manipule un vecteur d'entiers. Votre programme doit:

- Déclarer le tableau suivant :

```
int tab[] = {1,2,3,4,5,6,7,8,9,10};
```

- Définir ce qui suit :

- o 2 fonctions globales:

- **Carre** : elle permet de calculer le carré d'un entier passé par référence, et retourne le résultat.
- **Affiche** : elle permet d'afficher sur la sortie standard un entier, passé en argument.

- o La classe « **Additionne** » : suite à son appel à travers l'opérateur d'appel de fonction (), elle modifie la valeur d'un entier en lui additionnant le carré d'une valeur entière donnée (par défaut égale à 10), passée en paramètre. Le carré de cette valeur est calculé en utilisant la fonction carre précédemment définie.
- o La fonction principale « **main** » : elle utilise la classe et les fonctions globales pour manipuler le tableau « tab », en s'aidant dans cette tâche de la batterie d'algorithmes définis dans les STL.

- Afficher en sortie, après l'avoir exécuté, le résultat suivant:

```
Carré des 10 premiers entiers: 1 4 9 16 25 36 49 64 81 100
Additionne le carré de 100 à chaque élément du tableau :
10001 10004 10009 10016 10025 10036 10049 10064 10081 10100
Additionne le carré de la valeur par défaut à chaque élément du tableau :
10101 10104 10109 10116 10125 10136 10149 10164 10181 10200
```

- **Contrainte** : Dans cet exercice, il vous est interdit d'utiliser les structures de contrôle suivantes: if, while, for, do/while, switch/case, break, continue. Par ailleurs, il faudra respecter le nom du fichier.

Exercice 2 : Soit le conteneur « list » de la STL, contenant les noms de 3 fichiers (string) « t1.txt », « t2.txt » et « t3.txt ». Écrire le programme « tp4exo2.cpp » (et le « .h » associé si nécessaire) permettant de réaliser ce qui suit:

- Compter le nombre de caractères dans chacun des fichiers du conteneur « list ».
- Si un des fichiers du conteneur « list » est inexistant, un message d'erreur devra s'afficher à l'écran. Par exemple, si « t1.txt » n'existe pas : « impossible d'ouvrir le fichier: t1.txt », et par la suite, le programme devra poursuivre le traitement des autres fichiers.
- Le résultat doit être écrit dans le fichier « sortie.txt » sous le format suivant :

```

Nombre de fichiers dans le conteneur list: 3
Nombre de fichiers inexistant: 1
Nombre total de caractères: 810

Nom de fichier      : Nombre de caractères
t1.txt              : -1
t2.txt              : 560
t3.txt              : 250

```

Où les nombres « -1 », « 560 » et « 250 » représentent respectivement le nombre de caractères dans les fichiers « t1.txt », « t2.txt » et « t3.txt ». Pour un fichier inexistant (dans cet exemple « t1.txt »), il lui est associé la valeur -1. Le nombre « 810 » est la somme de « 560 » et « 250 ».

- **Contrainte :** Il faudra respecter le nom du fichier.

Exercice 3 : Expliquer les différentes instructions du programme et son fonctionnement. Répondre aussi aux différentes questions incluses dans le programme « tp4exo3.cpp ». Vos réponses doivent être formulées dans le fichier : « tp4exo3.pdf »

Hypothèses et contraintes : Celles déjà indiquées dans l'énoncé et celles par défaut déjà mentionnées dans les précédents travaux ou en cours (plagiat, version du compilateur, compiler avec l'option pedantic active, écrire un code en C++ et non pas C etc.).

Remise : Il est important de noter que votre TP sera compilé avec gcc3.2.4. Si par choix vous décidez d'utiliser un autre compilateur, vérifiez que le code que vous avez produit (qui normalement fonctionne correctement chez vous) fonctionne bien sur les ordinateurs de la DESI. Pour avoir la version du compilateur, utilisez la commande "gcc -v", qui devra donner le numéro de version "3.2.4".

Les fichiers à remettre :

- o **Version électronique :** Tous les fichiers associés aux exercices 1, 2 et 3.
- o **Version papier :** Juste la version papier de vos rapports.

Vous devez remettre les fichiers indiqués précédemment et cela avant le vendredi 28 avril 2006 à 09h00 :

1. Regrouper d'abord l'ensemble des fichiers à remettre dans un seul fichier « tp4.zip ».
2. Remettre par la suite ce fichier par la procédure de remise électronique habituelle (Pour obtenir de l'aide sur cette commande, tapez dans un Xterm : man remise). Respecter les noms des fichiers.
remise ift1169 tp4 tp4.zip
3. Vérifier que la remise s'est effectuée correctement.
remise -v ift1169 tp4
4. Remettez une **copie papier** de vos rapports sous la porte de mon bureau au 3225.

Barème : Ce TP4 est noté sur 12 points.

Exercice -1-	4	
Exercice -2-	4	
Exercice -3-	4	1 (Bonus)

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 papier vous pénalise de 2 points.
- Les programmes ne contenant pas d'en-tête, -2 points.
- Un programme qui ne compile pas : 0.
- Un programme qui compile mais ne fait 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.
- Le non respect du nom du fichier entraîne une erreur de compilation donc un des points de la spécification n'a pas été respecté : 0.
- 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 email, au moins la chaîne: [IFT1169] et une référence au **tp04**.

Mise à jour

08-04-2006 diffusion