

**IFT1169 – TRAVAIL PRATIQUE #3 – 24 février 2010****« Templates et STL »**

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 26 mars 2010, 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 d'une part les fonctions et les classes génériques, d'autre part quelques conteneurs de la librairie standard.

**Attention :** pour réaliser les exercices, vous devez vous baser sur les différentes fonctionnalités des conteneurs de la « STL » décrits dans les exercices. Vous devez aussi vous assurer d'avoir optimisé cette utilisation.

**Énoncé :** ce travail est composé de 3 exercices. Les étapes de compilation et d'édition de liens pour les 3 exercices doivent être regroupées dans un seul fichier « makefile ». Chaque cible dans le fichier correspond à un exercice.

Exercice	Cible
1a	exo01a
1b	exo01b
2	exo02
3	exo03

**Exercice -1-**

La classe générique « Tableau » dérive par héritage de la classe « Vector ». La classe « Tableau » en plus de supporter les opérations usuelles de la classe « Vector », permet aussi l'utilisation de l'opérateur de concaténation « + » et cela comme suit :

1	// Pour « exolatp3h10.cpp » et « exolbtp3h10.cpp »
2	
3	#include "tableau.h"
4	
5	int main() {
6	
7	Tableau<string> x(8), y(9), z;
8	z.push_back("Allo");
9	cout << (x+y+z).size() << endl; // Affiche 18
10	cout << (x+z)[8] << endl; // affiche Allo
11	return 0;
12	}

Afin de permettre de réaliser les opérations décrites dans les lignes 9 et 10 du précédent programme :

-1a- Écrire le contenu de la classe générique « Tableau » ainsi que l'opérateur « + » comme étant membre de la classe Tableau. (« tableau1a.h »)

-1b- Écrire le contenu de la classe générique « Tableau » ainsi que l'opérateur « + » comme étant non membre de la classe « Tableau ». (« tableau1b.h »)

**Exercice -2-**

Écrire le programme « exo02tp3h10.cpp » ayant la fonction « enlever » qui ne retourne rien et accepte deux arguments du type « string » : « phrase » et « sequence ». Le premier est passé par référence et représente la chaîne à modifier. Le second est une constante référence et représente le mot à rechercher dans la phrase. Le but de la fonction « enlever » est de rechercher dans la chaîne de caractères « phrase » la présence du mot « sequence » et de le retirer sauf la première occurrence du mot. Si après le retrait, une nouvelle séquence apparaît, elle est retirée elle aussi.

Phrase avant	Séquence	Phrase après
xx yy xxzx zy xzxzy xx	xx	xx yy zx zy xzxzy
12345 123456 345678	345	12345 126 678
xx y xxx yz xxw	xx	xx y x yz w

**Exercice -3-**

Soit un garage de voitures. Chaque voiture du garage a une couleur donnée et a été fabriquée par une entreprise donnée. Nous avons décidé de représenter ce garage par un conteneur « map ». La clé de ce conteneur est sous la forme d'une instance de voiture et sa valeur n'est qu'un vecteur contenant les noms des propriétaires des voitures ayant la même couleur et le même nom du fabricant.

Clé (Fabricant, Couleur)	Valeur
Honda, bleue	Felipe
Audi, noire	Marc
Toyota, grise	Bernard
Honda, bleue	Marie
Nissan, verte	Leila
Audi, noire	Momo

La classe « Voiture » est définie comme suit :

```
class Voiture {
public:
    Voiture(const string &m, const string &c) : Fabricant(m), Couleur(c) {}
    const string& getFabricant() const { return Fabricant; }
    const string& getCouleur() const { return Couleur; }
private:
    // REPRESENTATION
    string Fabricant;
    string Couleur;
};
```

Nous allons définir une fonction « affiche\_garage ». Ce dernière prend comme argument la précédente « map » et ne retourne rien. Elle permet d'afficher en sortie ce qui suit :

```
Les personnes qui conduisent une voiture Audi noire :
    Marc
    Momo
Les personnes qui conduisent une voiture Honda bleue :
    Felipe
    Marie
Les personnes qui conduisent une voiture Nissan verte :
    Leila
Les personnes qui conduisent une voiture Toyota grise :
    Bernard
```

À noter que le rapport affiché en sortie respecte un certain ordre : les fabricants par ordre alphabétique, ensuite par la couleur de la voiture, finalement les propriétaires de voitures identiques sont listés dans un ordre chronologique (l'ordre dans lequel ils ont parqué leur voiture dans le garage.)

-1- Pour que la classe « Voiture » décrite précédemment puisse être utilisée correctement par la « map », qu'avons-nous besoin de définir en plus, comme fonction non-membre de la classe « Voiture »?

-2- Écrire le code de la fonction « affiche\_garage ».

-3- Quand une personne récupère sa voiture, on la retire du garage. Écrire le code de la fonction « retire\_voiture » qui retourne un booléen, vrai pour signifier que la voiture a été retirée avec succès, sinon c'est faux. Cette fonction accepte les arguments suivants : une « map » sur les voitures passées par référence, le nom de la personne, la couleur de sa voiture et son fabricant.

-4- Regrouper les points 1 à 3 dans un programme complet « exo3tp3h10.cpp » avec une fonction « main » permettant de tester ces fonctionnalités.

**Fichiers à remettre :** vous devez remettre les fichiers nécessaires pour une compilation et une exécution correctes de vos programmes. Vous devez inclure aussi un fichier « makefile » permettant d'automatiser la compilation de vos fichiers. Par ailleurs, vous devez inclure dans votre remise un rapport décrivant le travail effectué. Ce rapport doit porter le nom « tp3H10.pdf ».

**Remise :** il est important de noter que votre TP sera compilé avec « gcc4.4.0 ». Si par choix vous décidez d'utiliser un autre compilateur, vérifiez que le code que vous avez produit (devant normalement fonctionner correctement chez vous) fonctionne aussi 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 « 4.4.0 ».

Par ailleurs, assurez-vous de la présence de l'option « pedantic » sur la ligne de compilation.

Pour éviter d'omettre un des fichiers associés à ce travail lors de la remise, nous vous conseillons de regrouper l'ensemble des fichiers dans un seul fichier compressé **tp3h10.zip** et de remettre ce dernier comme suit :

1. commencez d'abord par vous connecter sur la machine « **remise** » comme cela a été pratiqué dans la démo #01.
2. envoyez l'ensemble de vos fichiers par la procédure de remise électronique habituelle (Pour obtenir de l'aide sur cette commande, tapez dans un Xterm : man remise). Respectez les noms des fichiers.

**remise ift1169 1169tp3 tp3h10.zip**

**Attention: le nom du répertoire a changé par rapport aux tp#1 et tp#2. C'est « 1169tp3 ».**

3. vérifiez que la remise s'est effectuée correctement.

**remise -v ift1169 1169tp3**

**Barème :** ce TP3 est noté sur **13 points**. Les points sont répartis comme suit :

<b>Exercice 1</b>	<b>2.5 x 2 = 5</b>
<b>Exercice 2</b>	<b>3</b>
<b>Exercice 3</b>	<b>5</b>

**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.
- Les programmes ne contenant pas d'en-tête, -1 point.

- 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.
- Le non respect du nom du fichier va générer 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](mailto: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 **tp03**.

### **Mise à jour**

24-02-2010 diffusion de l'énoncé.