

IFT1166 – TRAVAIL PRATIQUE #1 – 13 septembre 2005

L'art de formater une sortie en C++!

Mohamed Lokbani

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

Remise : Deux remises à effectuer : électronique et papier le **mercredi 21 septembre, 21h00 au plus tard, sans possibilité de retard.**

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

But : Ce TP a pour but de vous faire pratiquer les outils nécessaires pour formater une sortie d'une part et d'autre part de vous permettre de vous familiariser avec votre environnement de travail. Et finalement vous faire pratiquer la procédure de remise, la rédaction d'un rapport et le respect des échéances.

Énoncé : On vous demande d'écrire dans un même programme C++, les instructions nécessaires permettant d'afficher en sortie des nombres dans divers formats. **Vous devez utiliser pour cela les manipulateurs appropriés¹.** Un manipulateur permet de préciser le format des données, quand il est inséré dans les instructions de lecture (entrée) et d'écriture (sortie) d'un flot. Une description détaillée de ces manipulateurs se trouve en annexe -1-. Pour utiliser ces manipulateurs, il faudra inclure dans le programme le fichier <iomanip>.

Description des affichages : Il vous est donc demandé d'écrire les instructions associées aux 10 affichages suivants (à faire dans l'ordre).

# Test	Nombres	Format demandé ainsi que les quelques particularités de l'affichage
1	99.5 31.275 77.0	Format par défaut + largeur minimale égale à 8 caractères (pas d'espace blanc entre les nombres).
2	512	Ce nombre est affiché 3 fois (pas d'espace blanc entre les nombres). Pour chaque cas : Format par défaut + largeur minimale égale à 8 caractères. Les 3 cas se distinguent que par la base utilisée: base 8, base 16 et la base courante. Vous utiliser pour cette question, le sélecteur de base.
3	512	Ce nombre est affiché 3 fois (pas d'espace blanc entre les nombres). Pour chaque cas : Format par défaut + largeur minimale égale à 8 caractères. Les 3 cas se distinguent que par le caractère de remplissage: '*', '0', ' ' (espace).
4	7.12345	Ce nombre est affiché 3 fois (pas d'espace blanc entre les nombres). Pour chaque cas : Format par défaut + largeur minimale égale à 8 caractères + représentation en virgule fixe. Les 3 cas se distinguent que par le nombre de caractères significatifs affichés (i.e. la précision): 2, 3 et 4.
5	512	Ce nombre est affiché 3 fois (pas d'espace blanc entre les nombres). Pour chaque cas : Format par défaut + largeur minimale égale à 8 caractères + justification à gauche.
6	512	Ce nombre est affiché 3 fois (pas d'espace blanc entre les nombres). Pour chaque cas : Format par défaut + largeur minimale égale à 8 caractères + justification à droite.
7	512	Format par défaut + largeur minimale égale à 8 caractères + son signe (ici +)
8	512.0	Format par défaut + largeur minimale égale à 8 caractères + forcer l'apparition du séparateur de virgule (ici le point).
9	512.0	Format par défaut + largeur minimale égale à 8 caractères + représentation scientifique.
10	7890	Format par défaut + affichage en base 16 + La base de numérotation utilisée ainsi que le chiffre doivent être indiqués en majuscule.

¹ Les solutions du genre un cout avec un paquet d'espaces blancs sont refusées.

Fichier de sortie : le résultat obtenu suite aux 10 affichages demandés, se trouve dans le fichier « **sortierf.txt** ». Il est disponible sur la page web du cours. Il faudra vous aider de ce fichier pour analyser la manière avec laquelle, les données doivent être affichées.

Hypothèses et contraintes :

- C'est un travail à faire **SEUL** ! Et seul ne signifie pas deux ni dix. Seul signifie aussi **réfléchir seul**. Plagiat équivaut à un 0 pour commencer.
- IFT1166 est un cours sur le C++, donc votre programme doit être écrit en C++ et pas en C ni en Java! Nous n'autoriserons aucune référence au langage C. Par exemple l'instruction [#include <stdio.h>] fait référence au langage C, donc non autorisée. Utiliser plutôt [#include <iostream>].
- Ce travail n'est pas un exercice d'algorithmique, donc pas besoin de compliquer le travail pour rien!
- Assurez vous de respecter le nom du fichier demandé, ainsi que le format de l'affichage en sortie.
- Le fichier de sortie a été généré sous **MinGW/Msys**. Il est donc au format DOS étendu. Si vous travaillez sur LINUX assurez-vous de les convertir au format LINUX en utilisant pour cela la commande **dos2unix**. Pour convertir de LINUX vers le format dos utiliser plutôt la commande **unix2dos**. Ces commandes sont disponibles dans un xterm d'une plateforme LINUX ou sous MinGW/Msys.

Remise : Il est important de noter que votre TP sera compilé avec gcc3.4.2. Si par choix vous décidez d'utiliser un autre compilateur ou une autre version de compilateur, vérifiez que le code que vous avez produit (qui normalement fonctionne correctement chez vous) fonctionne bien sur les ordinateurs du DESI. Il est de votre responsabilité de valider votre travail sur un des postes de la DESI avant de le remettre.

Pour avoir la version du compilateur utiliser la commande : "**gcc -v**" devra donner le numéro de version "**3.4.2**".

Vous devez remettre 2 fichiers : « **tp1.cpp** » et « **rapport** ». Le fichier « **tp1.cpp** » va contenir votre code alors que le fichier « **rapport** » va décrire comment vous avez procédé pour réaliser ce travail (pour le contenu d'un rapport voir la FAQ sur la page web du cours).

La remise comprend **3 (TROIS)** choses (avant le mercredi 21 septembre à 21h00) :

1. Envoyez vos fichiers « tp1.cpp » et le rapport 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 ift1166 tp1 tp1.cpp rapport

2. Vérifier que la remise s'est effectuée correctement.

remise -v ift1166 tp1

3. Remettez une **copie papier** de votre programme (« tp1.cpp ») ainsi que de votre rapport.

Barème : Ce TP1 est noté sur 4 points.

Compilation et respect des spécifications	0.5
Codage, commentaires etc.	1.25
Rapport	0.75
Tests (fourni)	1.5

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 1 point.
- Les programmes ne contenant pas d'en-tête, -1 point.
- Un programme qui ne compile pas : 0/4.

- Un programme qui compile mais ne fait les choses prévues dans la spécification : 0/4.
- Les avertissements (warnings) non corrigés : cela dépend de la quantité! À partir de -0.25 et plus.
- Le nom respect du nom de fichier, donc forcément il ne va pas compiler et un des points de la spécification n'a pas été respecté : 0/4.
- Aberration dans le codage : Même si tous les chemins mènent à Rome, faites l'effort nécessaire pour éviter de prendre le plus long!

Automatisation de la correction : C'est « l'ordinateur » qui va corriger en quelque sorte vos travaux. Ce dernier va utiliser des scripts qui vont tenir compte au détail près des contraintes imposées. Pas d'espaces blancs en trop, pas de commentaires supplémentaires. Vous devez produire une sortie **identique** à celle qui se trouve dans le fichier « **sortieref.txt** ».

Pour s'assurer de cela :

- 1- Commencer d'abord par rediriger la sortie dans un fichier avec le symbole >

```
tp1.exe > masortie.txt
```

- 2- Comparer par la suite votre sortie avec la sortie de référence en utilisant pour cela la commande diff (sous MinGW/MSys sinon la commande fc sous dos, ajuster juste les options) comme suit :

```
diff sortieref.txt masortie.txt
```

Si la commande « diff » ne signale pas de différences entre les deux fichiers, vous êtes sur la bonne voie pour obtenir une note élevée. Ainsi donc au moment de la remise de votre travail, vous avez déjà une idée approximative de la note que vous allez obtenir!

Des questions à propos de ce TP?

Une seule adresse : pift1166@iro.umontreal.ca

Pour faciliter le traitement de votre requête, inclure dans le sujet de votre email, au moins la chaîne: [IFT1166] et une référence au **tp1**.

Mise à jour

13-09-2005 Diffusion de l'énoncé.

Annexe²Manipulateurs de flux de sortie

Manipulateur	Fonction
endl	Envoie un caractère de retour à la ligne sur le flux et synchronise le tampon par un appel à la méthode flush.
ends	Envoie un caractère nul terminal de fin de ligne sur le flux.
flush	Synchronise le tampon utilisé par le flux par un appelle à la méthode flush.
boolalpha	Active le formatage des booléens sous forme textuelle.
noboolalpha	Désactive le formatage textuel des booléens.
hex	Formate les nombres en base 16.
oct	Formate les nombres en base 8.
dec	Formate les nombres en base 10.
fixed	Utilise la notation en virgule fixe pour les nombres à virgule.
scientific	Utilise la notation en virgule flottante pour les nombres à virgule.
left	Aligne les résultats à gauche.
right	Aligne les résultats à droite.
internal	Utilise le remplissage des champs avec des espaces complémentaires à une position fixe déterminée par la locale courante. Équivalent à right si la locale ne spécifie aucune position de remplissage particulière.
showbase	Indique la base de numérotation utilisée.
noshowbase	N'indique pas la base de numérotation utilisée.
showpoint	Utilise le séparateur de virgule dans les nombres à virgule, même si la partie fractionnaire est nulle.
noshowpoint	N'utilise le séparateur de virgule que si la partie fractionnaire des nombres à virgule flottante est significative.
showpos	Écrit systématiquement le signe des nombres, même s'ils sont positifs.
noshowpos	N'écrit le signe des nombres que s'ils sont négatifs.
uppercase	Écrit les exposants et les chiffres hexadécimaux en majuscule.
nouppercase	Écrit les exposants et les chiffres hexadécimaux en minuscule.
unitbuf	Effectue une opération de synchronisation du cache géré par le tampon du flux après chaque écriture.
nounitbuf	N'effectue les opérations de synchronisation du cache géré par le tampon du flux que lorsque cela est explicitement demandé.

Tableau -1- Manipulateurs sans paramètre

² Les tableaux 1 et 2 ont été extraits des notes de cours de C. Casteyde ©

Manipulateur	Fonction
<code>resetiosflags (ios_base::fmtflags)</code>	Permet d'effacer certains bits des options du flux. Ces bits sont spécifiés par une combinaison logique de constantes de type <code>ios_base::fmtflags</code> .
<code>setiosflags (ios_base::fmtflags)</code>	Permet de positionner certains bits des options du flux. Ces bits sont spécifiés par une combinaison logique de constantes de type <code>ios_base::fmtflags</code> .
<code>setbase (int base)</code>	Permet de sélectionner la base de numérotation utilisée. Les valeurs admissibles sont 8, 10 et 16 respectivement pour la base octale, la base décimale et la base hexadécimale.
<code>setprecision (int)</code>	Permet de spécifier la précision (nombre de caractères significatifs) des nombres formatés.
<code>setw (int)</code>	Permet de spécifier la largeur minimale du champ dans lequel la donnée suivante sera écrite à la prochaine opération d'écriture sur le flux.
<code>setfill (char_type)</code>	Permet de spécifier le caractère de remplissage à utiliser lorsque la largeur des champs est inférieure à la largeur minimale spécifiée dans les options de formatage.

Tableau -2- Manipulateurs avec paramètres

Remarques importantes

-1- Les manipulateurs de la table -1- ne sont pas tous du type « `ios::fmtflags` ». Les manipulateurs ayant ce type sont : { `boolalpha`; `dec`; `fixed`; `hex`; `internal`; `left`; `oct`; `right`; `scientific`; `showbase`; `showpoint`; `showpos`; `skipws`; `unitbuf`; `uppercase`; `adjustfield`; `basefield`; `floatfield` }.

-2- On peut activer (ou désactiver) plusieurs formats. Dans ce cas, séparer chaque format par l'opérateur « | » qui correspond au « ou logique » mais appliqué au niveau des bits. Par exemple:

```
« setiosflags (ios::unitbuf | ios::boolalpha); »
```

Ici le formatage correspond à effectuer une opération de synchronisation de la cache et l'activation du formatage textuel des booléens.

-3- On peut activer (ou désactiver) qu'un seul format dans ce cas pas besoin d'utiliser « | », par exemple : « `setiosflags (ios::unitbuf);` ».

-4- Si un manipulateur n'est pas invoqué, le flot utilise la valeur par défaut ou bien la dernière valeur connue pour ce manipulateur. De ce fait, faire très attention! Si on ne veut pas utiliser les dernières valeurs connues des manipulateurs, il devient nécessaire de les remettre à leur état initial.