

Date: Mon, 14 May 2007 13:29:19 -0400
From: Mohamed Nabil Lokbani <lokbani@iro.umontreal.ca>
To: liste IFT1166
Subject: [IFT1166] Tp#01 Q&R -14052007-

Bonjour,

Il y a 2 questions qui reviennent trop souvent:

=====

Q- est-ce qu'on doit afficher aussi les "extras" comme certaines phrases utilisées, les sauts de ligne, les quelques espaces blancs ici et la.

R- Oui, oui, oui, ouuuui. Sinon votre "diff" va faire des erreurs! Vous ne saurez pas si ces erreurs sont causées par ces "omissions" ou bien parce que vous avez un GROS bug vicieux caché quelque part dans votre affichage.

Certains m'ont signalé ce cas:

Dans la quatrième série, il y a un espace blanc en trop à la fin de la deuxième ligne : " 400 en base 8 vaut : 0620 ".

vous l'ajoutez à la main.

=====

Q- Est-ce qu'il n'y aurait pas une erreur avec l'affichage suivant:

Dans la dernière série

Un champ de 10 caracteres avec remplissage
e arrondi a 2 decimales : *****7.4
e arrondi a 4 decimales : *****7.389

Avec une précision de 2 chiffres ou 4 chiffres on ne devrait pas avoir plutôt ces sorties:

Un champ de 10 caracteres avec remplissage
e arrondi a 2 decimales : *****7.38
e arrondi a 4 decimales : *****7.3891

R- La réponse est non. L'affichage fourni est CORRECT. Pourquoi?

"setprecision" a une double fonction:

-1- si vous utilisez les options par défaut pour l'affichage en sortie (i.e. sans rien préciser), dans ce cas c'est l'état par défaut qui sera utilisé. Il n'est ni en virgule fixe ni en virgule flottante, c'est quelque chose entre les deux. Le "setprecision" va fixer dans ce cas le nombre de digits qui seront affichés.

Par exemple:

```
setprecision (3) << 7.1825
```

```
va donner: 7.18  
3 digits: 7 1 8
```

-2- si vous utilisez setprecision en utilisant comme manipulateur "la virgule fixe", setprecision va fixer dans ce cas le nombre de décimal à afficher après la virgule.

Par exemple:

fixez d'abord la sortie en virgule fixe, puis:

```
setprecision (3) << 7.1825
```

```
va donner: 7.182
```

donc 3 chiffres après la virgule.

-3- Il y'a un cas où il faudra faire gaffe! Si vous demandez à afficher par exemple 5 chiffres après la virgule et vous n'avez en réalité que 2 digits présents après la virgule. Il risque de compléter avec des 0. Même cas si vous demandez une précision de 20 et votre nombre ne peut avoir cette précision (sa représentation interne est limitée), il risque de compléter là aussi avec des 0.

Fin de l'histoire: Pourquoi vous obtenez des affichages différents de ceux fournis pour cet exemple?

Une règle d'or: ne pas oublier de remettre à zéro vos manipulateurs de sortie.

Comment avoir le coeur net? Il suffit de prendre votre instruction et la mettre dans un fichier à part et voir comment elle va se comporter. En faisant ainsi, vous assurez qu'aucun des manipulateurs utilisés dans les précédentes séries ne va pas mettre son grain de sel dans votre affichage.

Comment remettre à zéro ces manipulateurs? Pour l'exemple fourni en annexe -1-:

```
setiosflags(ios::noshowbase | ios::noboolalpha);
```

remise à zéro, j'utilise resetiosflags

```
resetiosflags(ios::noshowbase | ios::noboolalpha);
```

=====

Mohamed Lokbani.