Séance 1 - Révision /Prise en main de l'éditeur Emacs

A FAIRE SUR PAPIER

Exercice 1 (Expressions booléennes) Donner la table de vérité des expressions suivantes :

- non(a ou b)
- (a et non(b)) ou (b et non(a))
- non(a) et non(b)
- non(a ou b) ou (a et b)

Exercice 2 (Lecture de Code C++) Pour les programmes suivants, dire si ils sont corrects ou non. Si vous détectez des erreurs, les corriger. Ensuite, donner le résultat affiché.

```cpp
#include <iostream>

int main()
{
  int a,b;
  a= (1+2*3+8)/5+7;
  b=3*4*5-4;
  std::cout<<"a="<<a<<", b="<<b<<std::endl;
  return 0;
}
```

```cpp
#include <iostream>

int main() {
  int a,b;
  a=2;
  b=a++;
  std::cout<<"a="<<a<<", b="<<b<<std::endl;
  b+=a+2;
  a+=b--;
  std::cout<<"a="<<a<<", b="<<b<<std::endl;
  return 0;
}
```

```cpp
#include <iostream>

int main() {
  int a,b; float c;
  a=2; b=3; c=1.5;
  d=a+b+c;
  a=d;
  c+d=a-b;
  std::cout<<"a="<<a<<", b="<<b<<std::endl;
  std::cout<<"c="<<c<<", d="<<d<<std::endl;
  return 0;
}
```

```cpp
#include <iostream>
#include <iomanip>

int main() {
  float a,b,c;
  a=1.6;
  b=2*a+0.8;
  c=b/3;
  std::cout<<std::fixed;
  std::cout<<std::setprecision(2);
  std::cout<<"a="<<a<<std::endl;
  std::cout<<"b="<<b<<std::endl;
  std::cout<<"c="<<c<<std::endl;
  return 0;
}
```

Exercice 3 Soient les déclarations suivantes :

```cpp
int n=5, p=9;
int q;
float x;
```

Quelle est la valeur affectée aux différentes variables par chacune des instructions suivantes :

```cpp
q=n<p;
q=n==p;
q=p%n+p>n;
x=p/n;
x=(float)p/n;
x=(p+0.5)/n;
x=(int)(p+0.5)/n;
```

Exercice 4 (Nombres parfaits) Un nombre entier strictement positif est dit *parfait* s'il est égal à la somme de ses diviseurs propres + 1. Un diviseur propre d'un entier n est un diviseur de n différent de n et de 1.

Exemples : $6 = 2 \times 3$ et $6 = 2 + 3 + 1$;

$28 = 2 \times 14 = 4 \times 7$ et $28 = 2 + 14 + 4 + 7 + 1$;

Écrire un programme C++ qui affiche tous les nombres parfaits $\leq$ à une valeur donnée.

A FAIRE SUR MACHINE

Attention, certains documents sont à récupérer dans l'ENT. Pensez à vérifier que vous êtes inscrit sur le cours HLIN202 sur l'espace pédagogique si cela n'est pas déjà fait.

Consignes générales

Afin de travailler proprement, vous allez créer un répertoire HLIN202 à la racine de votre compte. Pour chaque TP vous créerez un nouveau répertoire, appelé TP1 (TP2, etc), dans le répertoire initial HLIN202. Vous écrirez les programmes associés à chaque TD-TP dans le répertoire correspondant. Pour faciliter votre travail, nous vous conseillons de nommer les fichier des programmes en référence au TD et à l'exercice concerné. Par exemple, le programme demandé à l'exercice 3 du TD/TP 1 sera nommé **td1-exo3.cpp**. Un bon façon de faire est également de prévoir un en-tête en début de fichier qui renseignes par exemple : auteur, date, numéro et but de l'exercice. Pour rappel, les commentaires en C++ se font par l'encadrement suivant : `/*` *commentaires* `*/`.

1 Création d'un programme avec EMACS

Pour créer un nouveau programme avec EMACS, il suffit de créer un nouveau fichier portant l'extension `.cpp`. Par exemple : `firstprog.cpp`. ATTENTION, pour nommer les fichiers (ainsi que les répertoires) en UNIX, veillez à ne pas utiliser de caractères spéciaux, de caractères accentuées ou d'espaces, limitez vous aux lettres majuscules, minuscules et aux tirets.

Pour créer un nouveau fichier avec EMACS, vous avez deux solutions :

— Dans un terminal, positionnez vous dans le répertoire ou vous voulez créer le fichier
`cd ~/HLIN202/TD1/` (par exemple)
Taper la commande : `emacs firstprog.cpp&` pour ouvrir EMACS et créer le fichier `firstprog.cpp`.

— Dans EMACS, sous le menu `File`, cliquez sur `"Open File ... "`. Tout en bas de la fenêtre EMACS vous trouverez le buffer interactif d'EMACS. Celui-ci affiche `"Find file: "` suivi du chemin du répertoire où vous avez ouvert EMACS ou bien du chemin du fichier courant dans EMACS. Il suffit de taper dans ce buffer le nom du fichier que vous souhaitez créer en le précédent du chemin complet dans l'arborescence des fichiers (rappel : le caractère `~` correspond au chemin associé au répertoire de votre compte). Ex : `"Find file: ~/HLIN202/TD1/firstprog.cpp "`. Valider en tapant sur la touche `Entrée`.

Si le fichier que vous essayer de créer existe déjà, alors ces deux solutions ne feront qu'ouvrir dans EMACS le fichier existant. Attention, penser à sauvegarder le fichier pour valider sa création. **Rappel :** Pour sauvegarder un fichier on clique dans le menu `File` puis `Save`. Sinon, en raccourci clavier on fait `<Ctrl-x><Ctrl-s>`.

2 Compilation d'un programme C++

Pour compiler un programme C++ il faut faire appel à un compilateur. Pour nos TP, nous utiliserons le compilateur C++ de la suite GNU Compiler Collection (GCC). Le compilateur est invoqué en ligne de commande en exécutant

```
g++ -Wall firstprog.cpp -o firstprog
```

Pour compiler vous avez 2 solutions :

— Ouvrez un terminal et positionnez vous dans le répertoire contenant votre programme et exécutez la commande de compilation `g++ -Wall firstprog.cpp -o firstprog`. Le compilateur affichera des erreurs de compilation si le programme n'est pas correct ou bien il générera l'exécutable de ce programme (dans notre exemple, `firstprog`).

— Dans EMACS, sous le menu `Tools`, cliquez sur `"Compile ... "`. Tout en bas de la fenêtre EMACS, le buffer interactif indique : `"Compile command: make -k "`. Positionnez vous dans le buffer et remplacez `"make -k "` par la commande de compilation `"g++ -Wall firstprog.cpp -o firstprog"`

et tapez sur la touche `Entrée` du clavier. Une sous-fenêtre apparaît dans EMACS pour donner le compte rendu de la compilation. Si aucune erreur n'est détecté, l'exécutable du programme sera généré (dans notre exemple, `firstprog`).

Attention, pensez bien à remplacer le fichier `firstprog.cpp` et le nom d'exécutable `firstprog` dans la commande de compilation par les noms correspondants à votre programme.

3 Exécution d'un programme

Dans un terminal, positionnez vous dans le répertoire où se trouve le programme puis lancez le en exécutant la ligne de commande suivante :

```
./firstprog
```

où `firstprog` correspond au nom que vous avez donné à votre programme. Attention, le `./` est obligatoire.

Remarque : vous pouvez également lancer le programme directement dans EMACS. Pour cela, dans le menu `Tools` cliquez sur `"Shell Command... "`. Tout en bas de la fenêtre EMACS, le buffer interactif indique : `"Shell command: "`. Tapez dans ce buffer la ligne de commande relative à l'exécution de votre programme (Ex : `"./firstprog "`) puis tapez sur `Entrée`. Une sous-fenêtre apparaît dans EMACS. Cette sous-fenêtre permettra de voir les affichages du programme. Attention, ce mode d'exécution ne permet pas d'interagir en entrée avec le programme (saisie de valeurs au clavier). Pour faire cela, vous devrez lancer le mode Shell interactif en tapant la combinaison de touche `ALT-x` puis entrée `shell` dans l'invite de commande d'Emacs.

4 Prise en main

4.1 Compilation et exécution

Dans un premier temps, nous allons essayer de compiler et exécuter un programme simple. Récupérez le programme `firstprog.cpp` disponible sur l'ENT (`Espace Pédagogique Claroline > HLIN202 > Documents et liens >` dans le répertoire `TP`. Enregistrez ce fichier dans le répertoire `HLIN202/TD1` que vous aurez pris la peine de créer. Compilez ce programme et exécutez le. Votre terminal ou EMACS doit afficher un message.

4.2 Compilation et correction d'erreurs

Récupérez sur l'ENT le programme `secondprog.cpp` et enregistrez le dans le même répertoire que précédemment. Compilez ce programme. Si vous avez des messages d'erreur c'est normal. Il vous faut ouvrir ce fichier avec EMACS et corriger une erreur. Compilez à nouveau ce programme et exécutez le.

4.3 Création d'un nouveau programme

Créer un nouveau fichier `thirdprog.cpp` avec EMACS. Écrire dans ce fichier le code d'un programme permettant d'afficher la valeur de l'expression suivante : `(1+2)*5+8/3-47`. Compilez et exécutez ce programme. **Vous êtes maintenant prêts pour commencer les exercices.**

5 Exercices

Exercice 5 Écrivez un programme qui demande de saisir 2 entiers au clavier et qui affiche leur somme et leur produit à l'écran.

Exercice 6 Écrivez un programme qui demande de saisir 2 entiers au clavier et qui les stocke dans 2 variables (a et b par exemple). Affichez à l'écran l'adresse des variables a et b ainsi que leur valeur. Ensuite, échangez les valeurs de a et de b et refaites le même affichage (adresse et valeur).

Exercice 7 Écrivez un programme qui définit dans une variable réelle une température exprimée en degré Fahrenheit et qui affiche à l'écran la conversion de cette température en degré Celcius.

$$C = \frac{5}{9}(F - 32)$$

Écrivez ensuite le programme qui fait l'opération inverse.
Il se peut que votre programme ne calcule pas le bon résultat (toujours 0). Si tel est le cas, pensez à vérifier l'expression représentant votre calcul en gardant en tête que le calcul avec des entiers renvoie un entier.