Licence Informatique **UE INF101**
Licence Mathématiques **Initiation la programmation**
Université Bordeaux 1 **Travaux Dirigés**
2006-2007 **feuille 1**

1 Travaux Dirigés

Exercice 1. On considère le programme `bonjour` dont le code suit :

```
#include <stdio.h>

int
main(void)
{
  printf("Bonjour␣!\n");
}
```

– Qu'affiche l'exécution de ce programme ?
– Comment le compiler, l'exécuter ?
– Pourquoi ce programme comporte-t-il la ligne `#include <stdio.h>` ? Quel est ce fichier ?
– Modifier le code du programme précédent pour obtenir l'affichage

```
Bonjour.
Comment ca va ?
```

Exercice 2. On considère le programme suivant :

```
#include <stdio.h>
#include "inf101.h"

int
main(void)
{
  int x = lire_entier();
  if (x == 1)
    printf("Bonjour␣!\n");
  else
    printf("Hello␣!\n");
}
```

1. Quel est l'affichage produit dans les cas suivants :

   la valeur retournée par `lire_entier()` est 1
   la valeur retournée par `lire_entier()` est 2
   la valeur retournée par `lire_entier()` est 10

2. Modifier le programme précédent pour afficher

   Bonjour ! avec 1, Hello ! avec 2, ne rien afficher avec d'autres valeurs.

Exercice 3.

1. Ecrire un programme qui calcule puis affiche le maximum de deux entiers.
2. À l'aide de la fonction définie à la question précédente, écrire un programme qui calcule et affiche le maximum de trois entiers.

Exercice 4. On considère le code suivant :

```
#include <stdio.h>
#include "inf101.h"
int main(void)
{
  int n = lire_entier();
  for (int x = 0; x < n ; x++)
    printf("%d␣*␣%d␣=␣%d\n", x, x, x*x);
}
```

Quel est le résultat de l'exécution de ce programme lorsque la valeur retournée par `lire_entier()` est 10 ?

Exercice 5. Écrire un programme `echobonjour.c` qui affiche plusieurs fois Bonjour !
Le nombre de fois est saisi (on utilisera la fonction `lire_entier()`).

Exercice 6. Faire tourner le programme `puissance_de_2` pour chacun des deux cas

la valeur retournée par `lire_entier()` est 0
la valeur retournée par `lire_entier()` est 3

```
#include <stdio.h>
#include "inf101.h"

int
puissance_de_2(int n)
{
  int p2 = 1;

  for (int i = 0; i < n; i++)
  {
    p2 = 2 * p2;
  }
  return p2;
}

int
main(void)
{
  int n = lire_entier();

  printf("%d\n", puissance_de_2(n));
}
```

Exercice 7. En vous inspirant du code précédent, écrire le code `puissances.c` tel que `./puissances` affiche

```
64, 32, 16, 8, 4, 2, 1
```

Exercice 8. Que fait le code suivant ? Réécrire ce code correctement.
#include<stdio.h>
int main(void){for(int compteur=0 ;compteur<10 ;compteur++){int truc=2*compteur+1 ; printf("%d\n",truc) ;}}

Exercice Complémentaire 1. Ecrire un programme C, `une_minute_en_plus`, qui affiche l'heure une minute après celle lue sous forme de deux entiers saisis avec `lire_entier()`.

2 Travaux Pratiques

Exercice 9.

PRÉPARATION

Rappel : pour obtenir une documentation en ligne sur une commande unix, on utilise la commande `man` (taper **q** pour sortir de la page de manuel).
Exemple : `man less` pour l'aide sur la commande **less**

1. À votre racine créer le répertoire `initiation-programmation`
   ```
   mkdir initiation-programmation
   ```
   Ce répertoire contiendra l'ensemble de votre travail en initiation à la programmation ;
2. Se placer dans le répertoire `initiation-programmation`
   ```
   cd initiation-programmation
   ```
   Y créer le sous-répertoire `src`.
   Se placer dans le répertoire `src`.
3. Télécharger les fichiers qui accompagnent le Tp à l'aide de la commande
   `wget http://dept-info.labri.fr/ENSEIGNEMENT/InitProg/src/fichiers-1.tar.gz`
4. Décompacter cette archive au moyen de la commande `tar -xvzf fichiers-1.tar.gz`
   Le répertoire td01 est créé automatiquement.
5. Visiter succintement le répertoire `initiation-programmation/src/td01` et son contenu à l'aide des commandes `ls` et `less`

ÉDITION D'UN FICHIER

6. Lancer l'exécution de l'éditeur **emacs**[1] en arrière-plan en utilisant la ligne de commande :
   ```
   emacs &
   ```
7. Sous `emacs` charger le fichier `bonjour.c` en tapant
   `C-x C-f ~/initiation-programmation/src/td01/bonjour.c` [2]
   Il est conseillé d'utiliser la complétion automatique (touche `<TAB>`) pour la saisie du nom de fichier.

COMPILATION

8. Sous un shell, se placer dans le répertoire `~/initiation-programmation/src/td01`, lister son contenu, puis exécuter les commandes suivantes (on analysera l'effet de chacune des commandes) :
   ```
   gcc -Wall -std=c99  -Werror bonjour.c -o bonjour
   ls
   ./bonjour
   rm bonjour
   ls
   ```
9. Compiler sous `emacs` le programme `bonjour.c` ; Pour cela taper
   `M-x compile` [3]
   puis
   remplacer `make -k` par
   ```
   gcc -Wall -std=c99 -Werror bonjour.c -o bonjour
   ```
10. Dans le shell, vérifier le contenu du répertoire **td01** et exécuter le programme `bonjour`.

[1] il est préférable de ne pas lancer plusieurs processus emacs en parallèle
[2] C-x C-f se dit "contrôle x, contrôle f" et signifie appuyer simultanément sur la touche "Ctrl" et sur la touche "x" puis appuyer simultanément sur "Ctrl" et "f"
[3] Pour otenir M-x (méta x), il faut appuyer successivement sur les touches "Esc" puis x.

Exercice 10. Recommencer l'exercice précédent avec `puissance_de_2.c`.
Afin d'inclure dans le code la définiton de la fonction `lire_entier()` définie dans le fichier `inf101.c`, on compilera en utilisant :

```
gcc -Wall -std=c99 -Werror inf101.c puissance_de_2.c -o puissance_de_2
```

Exercice 11.

1. Editer avec emacs le code `mal_ecrit.c`
2. Corriger et indenter correctement ce code.
3. Sauver ce code sous le nom `bien_ecrit.c` avec la clé `C-x C-w`
   `C-x C-w ~/initiation-programmation/src/td01/bien_ecrit.c`
4. Lancer la compilation du programme depuis `emacs` ;
   Traiter les erreurs (éventuelles) de syntaxe en utilisant la clé `C-x ‘` [4]

Exercice 12. Éditer le programme `vol.c`. La compilation de ce programme produit des erreurs. En utilisant les messages d'erreurs produits par le compilateur, corriger le source de ce programme. Exécuter ce programme et en analyser le source. Quel est le résultat de son exécution ?

Exercice 13. Éditer, compiler, exécuter un programme `median` qui retourne le médian des trois entiers saisis à l'aide de la fonction `lire_entier()`

Exercice Complémentaire 2. Éditer, compiler, exécuter le programme `puissances` de la partie Travaux Dirigés.

[4] pour obtenir le symbole ‘ sur un clavier AZERTY, il faut taper "AltGr"-7 pour se positionner automatiquement dans le fichier contenant la première erreur et sur la ligne de l'erreur. Chaque nouvelle frappe de la clé **C-x ‘** provoque le positionnement sur l'erreur suivante, et ceci jusqu'à ce que toutes les erreurs aient été balayées.