Licence 1 - section B
TP 1 d'éléments d'informatique

Catherine RECANATI – Département d'Informatique – Institut Galilée

Semaine du 7 au 11 novembre 2016

1 Premiers pas sous Shell Linux et langage C

1. Depuis votre répertoire principal (base de votre arborescence de fichiers), créez le répertoire TP1 en tapant la commande `mkdir TP1` dans une fenêtre de terminal.
2. Allez dans le répertoire TP1 grâce à la commande `cd` (= *change directory*) en tapant : `cd TP1`.
3. Lancez l'éditeur de texte `gedit` pour créer un nouveau fichier source appelé `bonjour.c`, en tapant la commande `gedit bonjour.c &`. Le caractère & en fin de commande permet de lancer l'exécution de la commande en arrière plan (et ainsi garder la main dans la fenêtre du terminal, sans avoir à attendre la fin de la commande, ici donc sans attendre la fin de l'éditeur `gedit`). L'intérêt ici, sera de pouvoir lancer la compilation du programme et son exécution sans avoir à quitter l'éditeur `gedit`.

Exercice 1.1 Programme Bonjour !

Ce premier programme devra afficher Bonjour ! Vous le composerez en recopiant le programme du cours ou un programme donné en exemple en TD. Votre éditeur `gedit` vous assistera en coloriant automatiquement les mots du code saisi.

Pour réaliser l'affichage de Bonjour !, vous utiliserez l'instruction `printf("Bonjour !\n");` Le `\n` représente l'impression d'un saut à la ligne après l'impression de Bonjour ! `printf` est une fonction définie dans la bibliothèque standard d'entrée/sortie du C, et pour que le compilateur trouve cette définition, il faut insérer la ligne suivante au début de votre programme :

```
#include <stdio.h>        /* pour pouvoir utiliser printf */
```

1. Après avoir fini d'écrire votre programme, enregistrez-le.
2. Pour créer un programme exécutable à partir de votre fichier source, vous devez *compiler* votre fichier avec la commande `gcc`, qu'on détaillera au prochain cours :
   `gcc -Wall bonjour.c -o bonjour.exe`
   Si le compilateur vous signale des erreurs, corrigez-les dans l'éditeur, enregistrez et recompilez.
3. Quand l'étape précédente ne vous signale plus d'erreurs, lancez l'exécution du programme en tapant simplement le nom de son fichier code exécutable :
   `bonjour.exe` (ou `./bonjour.exe`).

Modifiez le programme de manière à ce qu'il affiche votre prénom après «Bonjour !». Vous répéterez ces trois étapes (écrire/sauvegarder, compiler puis exécuter), dans tous les TP.

Exercez-vous à apprendre les raccourcis clavier des diffrentes commandes de l'éditeur pour éviter d'utiliser la souris qui est moins rapide.

Exercice 1.2 Affichages

1. Écrire un programme `coucou.c` qui affiche à l'écran « Coucou ».
2. Modifier ce programme pour qu'il affiche à l'écran « Coucou » sur cinq lignes de deux façons :
   — avec cinq printf ;
   — avec un seul printf.
3. Modifier ce programme pour qu'il affiche à l'écran l'évaluation de l'expression `7 * 3 + 2`.
4. Modifier à nouveau ce programme pour qu'il affiche à l'écran l'évaluation de l'expression `3 * x + 2`, avec la variable entière `x` initialisée à une valeur quelconque dans le programme.

2 Lire et imprimer des variables (vu en TD)

On a vu en cours la fonction printf qui utilise des indications de formats avec le symbole %, comme par exemple %d indiquant un format d'impression décimal. Il existe d'autres formats reconnus par printf comme le format %x des nombres hexadécimaux. Le tableau suivant indiquent divers formats que vous pourrez tester dans un programme :

Formats	Sorties	Exemples
%d ou %i	Entier decimal signe	392
%u	Entier decimal non signe	7235
%o	Octal non signe	610
%x	Entier hexadecimal non signe	7fa
%X	Entier hexadecimal non signe	7FA
%f	Decimal virgule flottante, minuscules	392.65
%F	Decimal virgule flottante, majuscules	392.65
%e	Notation scientifique (mantisse/exposant min)	3.9265e+2
%E	Notation scientifique (mantisse/exposant maj)	3.9265E+2
%g	Utilise la representation la + courte: %e ou %f	392.65
%G	Utilise la representation la + courte: %E ou %F	392.65
%c	Caractere (char)	a
%s	Chaine de caracteres	bonjour
%p	Adresse (pointeur)	b8000000

A noter que pour les flottants, on peut indiquer le nombre de caractères figurant avant et après la virgule, comme avec `%3.1f` pour obtenir l'affichage 392.6 au lieu de 392.65.

La fonction qui permet de lire des caractères du clavier pour initialiser une variable s'appelle `scanf`. Elle prend (comme `printf`) en premier argument une chaine de caratères contenant des indications de format, et en second argument l'adresse en mémoire d'une variable. Pour connaitre l'adresse d'une variable, on lui appliquera l'opérateur & (prononcer "et commercial" ou "esperluette").

Ainsi, les lignes de programme suivantes permettent de lire une variable initialisée d'abord à zéro à partir des caractères entrés au clavier par l'utilisateur du programme :

```
1    int variable = 0;
2    scanf ("%d", &variable);   /* & est l'operateur d'adresse */
3    printf ("%d\n", variable);
```

La plupart des formats d'impression de printf du tableau précédent restent valables comme format de lecture pour `scanf`. Ainsi on pourra utiliser `%d %i` et `%u` pour lire des décimaux,`%x` et `%X` pour des hexadécimaux, `%o` pour des nombres en notation octale, `%e` , `%f` , et `%g` pour lire des float, et `%lf` pour des double (le l devant le f signifie long). Enfin `%c` et `%s` permettront de lire respectivement un caractère et une chaine de caractères.

A noter que si l'on préfixe l'indicateur de format d'un nombre, cela précisera le nombre de caractères qui seront lus par `scanf`.

```
1   int nb_de_5_chiffres = 0;
2   scanf ("%5d", &nb_de_5_chiffres);
3   printf ("---> %d\n", nb_de_5_chiffres);
```

Exécutions :

```
18
---> 18
1000
---> 1000
1584669842
---> 15846
```

Mais attention, si l'utilisateur tape plus de caracères que nécessaire, les caracères supplémentaires ne seront pas lus mais resteront dans le tampon qui lit les entrées. C'est eux qui se présenteront ensuite pour être lus si on fait un nouvel appel à la fonction scanf.

Exercice 2.1 Lire et imprimer une variable.

1. Ecrire un programme dont la fonction principale lit une variable x de type unsigned long int avec scanf et l'imprime ensuite dans les formats %lu, %lo et %lX.
2. Modifier le programme précdent pour que la fonction main appelle une fonction imprime(x) qui imprime son argument dans les formats %lu, %lo et %lX.

3 Quelques petits programmes

Exercice 3.1 Conversion Fahrenheit-Celsius.

Écrire un programme qui demande à l'utilisateur d'entrer une température en degrés Fahrenheit et affiche sa conversion en degrés Celsius. On s'appuiera sur la formule `[C] = ([F] - 32) x 5/9`.

Exercice 3.2 Majeur ou mineur ?

Écrire un programme qui demande à l'utilisateur d'entrer son âge et affiche ensuite si la personne est majeure ou mineure. L'algorithme devra d'abord être écrit en français en mettant l'indentation.

Exercice 3.3 Année bissextile ?

Écrire un programme qui demande à l'utilisateur d'entrer une année et qui affiche ensuite si l'année est bissextile ou non. On rappelle qu'une année est bissextile dans les deux cas suivants :

1. si l'année est divisible par 4 et non divisible par 100, ou
2. si l'année est divisible par 400.

Donc en particulier l'année 1900 n'est pas bissextile. Par contre l'an 2000 l'est.

Exercice 3.4 Minimum de 3 valeurs.

Soient 3 variables `a`, `b`, `c`, initialisées au clavier par l'utilisateur. Écrire une fonction minimum (a, b, c) qui calcule et retourne le minimum de ces 3 valeurs. Appeler la fonction dans le programme principal et imprimer le résultat.