
Travaux pratiques 2 et 3 : premiers pas en langage C, structure de contrôle *if* et itération *for*

Vous allez mettre tous vos programmes écrits dans ce TP dans le répertoire TP2 :

1 Shell et gcc

1. À partir du début de votre arborescence, créez le répertoire TP2 : `mkdir TP2`
2. Allez dans ce répertoire pour y mettre des fichiers : `cd TP2`
3. Créez un nouveau fichier source pour le langage C : `gedit bonjour.c &`

Ce premier programme devra afficher **Bonjour !**, vous le composerez en recopiant un programme donné en exemple dans un TD. Gardez tous les commentaires vous en aurez besoin plus tard. Votre éditeur de texte vous assistera en colorisant automatiquement le code saisi.

Pour réaliser l'affichage vous utiliserez l'instruction :

`printf("Bonjour !\n");`, dans votre fonction principale. Le `printf` est une *fonctionnalité supplémentaire* d'entrée sortie, pour qu'il fonctionne il faut insérer la ligne suivante au début de votre programme (après la ligne 2) :

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

Le `\n` représente un saut de ligne.

4. Après avoir fini d'écrire votre programme, enregistrez le.
5. Créez un programme exécutable à partir de votre fichier source, cette étape, dite de compilation, sera expliquée au prochain cours : `gcc -Wall bonjour.c -o bonjour.exe`
6. Quand l'étape précédente a réussi, exécutez le programme pour vérifier qu'il fonctionne : `bonjour.exe` (ou `./bonjour.exe`).

Vous répétez ces trois dernières étapes (écrire/sauvegarder/compiler exécuter), très souvent ce semestre et le suivant. Exercez-vous à tout faire avec les raccourcis clavier *sans utiliser la souris*.

7. Modifiez le programme de manière à ce qu'il affiche votre prénom après le «Bonjour» (compiler/exécuter).

2 Affichage

1. Écrire un programme `exo1.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. Écrire un programme `exo2.c` qui affiche à l'écran l'évaluation de l'expression $7 * 3 + 2$.
4. Modifier 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.

3 Itération *for* : Affichage de figures géométriques

Les exercices suivants utilisent le caractère * (étoile) pour dessiner des figures géométriques simples, appelées figures d'étoiles.

3.1 Exercice type : affichage d'un rectangle d'étoiles

Écrire un programme qui, étant données deux variables, `longueur` et `largeur`, initialisées à des valeurs strictement positives quelconques, affiche un rectangle d'étoiles ayant pour longueur `longueur` étoiles et largeur `largeur` étoiles. Deux exemples d'exécution, avec deux initialisations différentes, sont les suivants :

Affichage d'un rectangle d'étoiles de longueur 10 et largeur 5.

```
*****
*****
*****
*****
*****
```

Affichage d'un rectangle d'étoiles de longueur 6 et largeur 3.

```
*****
*****
*****
```

3.2 Exercice type : affichage d'un demi-carré d'étoiles

Écrire un programme qui affiche, étant donnée la variable, `cote`, initialisée à une valeur quelconque, un demi-carré d'étoiles (triangle rectangle isocèle) ayant pour longueur de côté `cote` étoiles. Deux exemples d'exécution, avec deux initialisations différentes, sont les suivants :

Affichage d'un demi-carre d'étoiles de cote 6.

```
*
**
***
****
*****
*****
```

Affichage d'un demi-carre d'étoiles de cote 2.

```
*
**
```

3.3 Affichage d'un demi-carré droit d'étoiles

Écrire un programme qui affiche un demi-carré droit d'étoiles de côté spécifié par l'utilisateur. Exemple d'exécution :

Entrer la taille du demi-carré :

5

Affichage d'un demi-carre droit d'étoiles de cote 5.

```
*
**
***
****
*****
```

4 Boucle de répétition while

Pour contrôler qu'une valeur demandée est bien dans un certain intervalle (ou domaine), on souhaite faire une boucle de répétition : on redemande la valeur tant qu'elle n'est pas dans cet intervalle. Par exemple on veut demander une valeur entière mais positive.

1. Ecrire un petit programme pour vérifier que si on demande une valeur entière elle peut-être négative. Vous utiliserez pour cela la fonction `scanf` qui permet de faire entrer dans le programme une donnée saisie par l'utilisateur. Par exemple, pour mettre dans une variable de type `int` une valeur saisie par l'utilisateur on peut écrire

```
scanf("%d", &x)
```

Le premier paramètre de `scanf` joue le même rôle que dans `printf` : il spécifie le format des données à lire (ou à écrire dans `printf`). En particulier, le symbol `%d` est le même déjà utilisé dans `printf` pour spécifier que la donnée correspondante est un entier (qui devra être affiché dans `printf` ou lu dans `scanf`). On remarque le symbol `&` juste avant le nom de la variable où on veut stocker la valeur lue. Ce symbol est indispensable, sans cela le programme ne se comportera pas correctement. Attention, la commande `scanf` n'affiche aucun message de requête avant de chercher de lire une valeur, il faudra donc utiliser `printf` pour pouvoir envoyer un tel message. Une fois lu l'entrée, le programme devra enfin afficher un message de confirmation contenant la valeur lue. (Toute l'interaction entrée/sortie contrôlée par `scanf/printf` se déroule sur le terminal utilisé pour exécuter le programme.)

2. Modifiez votre programme pour contrôler la valeur qui est saisie : vous utiliserez une boucle pour répéter la demande de la valeur entière tant que la saisie est négative. Enfin afficher un message indiquant que la valeur a été contrôlée. Pour cette boucle vous pouvez utiliser la boucle `while`, une variante de la commande `for` vue en cours et TD. Par rapport à la boucle `for`, le `while` n'a pas la partie d'initialisation et la partie de mise à jour des paramètres de contrôle de la boucle. Sa structure est la suivante

```
while ( * condition * )
{
    ...
    * bloc *
    ...
}
```

et `sont bloc` est évalué tant que la condition est vraie. En d'autres termes, un `while` correspond à un `for`

```
for ( ; * condition * ; )
{
    ...
    * bloc *
    ...
}
```

où la commande d'initialisation est mise à jour sont vides.