

1 Quelques rappels

Voici le squelette basique d'un programme en C :

```
#include "ma_bibli.h" // ma bibliotheque
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char* argv[]){
    // ceci est la fonction principale
    // instructions;
}
```

Nous parlerons de la première ligne dans le dernier exercice de ce TP.

Les deux lignes suivantes contiennent les bibliothèques “classiques” qu’il faut toujours inclure dans vos programmes en C.

La fonction “main” est, comme son nom l’indique, la fonction principale de votre programme, c’est elle qui est appelée lors de l’exécution de votre programme.

Voici comment compiler puis exécuter votre programme :

```
$ gcc -Wall mon_programme.c -o executable
$ ./executable
```

Question 1 : Créez un programme qui affiche “Hello World!” à l’écran.

Question 2 : Ajouter à votre programme précédent une fonction *moyenne* qui prend en paramètres deux entiers a et b et qui retourne un float de la moyenne entre a et b .

Question 3 : Modifiez votre programme de manière à ce qu’il affiche :

```
$ Hello World!
$ La moyenne entre a=10 et b=13 est 11.5
```

Naturellement vous pouvez choisir les nombres que vous souhaitez pour a et b .

2 Les différentes bases

En informatique, les bases souvent utilisées sont la base 16, dite hexadécimale, la base 2, dite binaire et enfin (plus rarement) la base 8, dite octale.

Question 1 : Créez un programme qui affiche un nombre a dans les 3 bases suivantes : décimale, hexadécimale et octale.

Astuce : il suffit d'utiliser les bons caractères de formatage de la fonction `printf`.

Question 2 : Créez une fonction qui affiche un nombre a sous forme binaire (avec le bit de poids fort à gauche), par exemple 6 sera affiché 1 1 0.

Question 3 : Modifiez votre programme pour qu'il affiche désormais le nombre a dans les 3 bases : 10, 16 et 2.

Question 4 : Modifiez votre programme pour que l'affichage soit parfaitement aligné, et pour mettre en valeur la différence entre le binaire et l'héxadécimal. Vous obtiendrez quelque chose de similaire à la figure 1.

```
nombre = 254
hexadecimal : fe
binaire :      1 1 1 1 1 1 1 0

nombre = 255
hexadecimal : ff
binaire :      1 1 1 1 1 1 1 1
```

FIGURE 1 –

Astuce : demandez à votre programme d'afficher tous les nombres entre 10 et 30 par exemple.

Question 5 (difficile) : Que remarquez vous entre l'écriture hexadécimale et binaire d'un nombre? Codez une fonction qui permet de passer d'une écriture hexadécimale, que vous insérerez au préalable dans une chaîne de caractères, à une écriture binaire.

Astuce : Je vous conseille d'utiliser une cousine de la fonction `printf`...

3 Un bon réflexe : l'utilisation de bibliothèques

Quand un programme contient un grand nombre de fonctions, il est vivement recommandé de les mettre dans des fichiers séparés, on parle de bibliothèques de fonctions. Par exemple `stdio` et `math` sont des bibliothèques très souvent utilisées.

À la différence de ces bibliothèques qui sont déjà compilées, vous allez devoir compiler les vôtres et elles seront automatiquement liées à votre programme à ce moment là.

Voici de manière synthétique la manière de procéder pour avoir une bibliothèque de fonctions, dite locale :

- mettez vos fonctions dans un fichier `ma_bibliotheque.c`
- créez le fichier `ma_bibliotheque.h` qui contient **uniquement** les prototypes de vos fonctions
- ajoutez la ligne `#include "ma_bibliotheque.h"` au début de votre fichier principal `mon_programme.c`

Voici comment compiler puis exécuter votre programme, **dans le cas où ce dernier utilise les fonctions d'une bibliothèque** :

```
$ gcc -Wall mon_programme.c ma_bibliotheque.c -o executable  
$ ./executable
```

Question 1 : Rappelez ce qu'est le prototype d'une fonction.

Question 2 : Créez une bibliothèque de fonctions contenant deux fonctions : la fonction de conversion d'un entier en écriture binaire (exercice 2) et la fonction qui affiche la moyenne de 2 nombres (exercice 1).

Question 3 : Créez un programme qui utilise la bibliothèque précédente pour afficher la moyenne de deux nombres (castée en entier), sous forme binaire. Compilez et exécutez ce code.

Vous devriez avoir un affichage de ce type :

```
$ La moyenne entre a=10 et b=13 est 1 0 1 1
```