

TP n°2

Programmation C (révisions)

Partie 1 : compilation et fonctions

```
#include <stdio.h>
int main(void) {
    printf("Hello World!!!\n");
    return 0;}
```

Ce programme est sauvegardé dans le fichier `hello.c`. Pour créer un fichier exécutable, le plus simple consiste en la commande :

```
$ gcc hello.c
```

Cette commande effectue l'ensemble des phases de la compilation et crée le fichier exécutable `a.out`. Avec l'option `-o` nous pouvons spécifier le nom du fichier exécutable :

```
$ gcc -o hello hello.c
```

Ici le programme exécutable est `hello`. Une autre possibilité, utile lors de compilation séparée, est de passer par l'intermédiaire de fichiers objets (`*.o`) :

```
$ gcc -c hello.c
$ gcc -o hello hello.o
```

L'option `-Wall` permet d'afficher tous les messages de warnings. Il est recommandé de prendre en compte ces messages afin d'avoir les bonnes habitudes de programmation en C.

```
$ gcc -Wall -o hello hello.c
```

Exercice 1

Écrire un programme qui saisit 2 entiers et affiche successivement la somme, la différence, le produit, les quotients de la division réelle et euclidienne et le reste de la division euclidienne de ces 2 entiers.

Exercice 2

Écrire deux programmes retournant la factorielle, calculée respectivement de façon itérative et récursive, d'un nombre entier entré par l'utilisateur.

Partie 2 : fonction, allocation de mémoire et pointeur

Exercice 3

Soit P un pointeur qui 'pointe' sur un tableau A :

```
int A[] = {12, 23, 34, 45, 56, 67, 78, 89, 90};  
int *P;  
P = A;
```

Quelles valeurs ou adresses fournissent ces expressions :

1. P
2. &A