

# TD1 de Programmation C

L1 MIASHS, 2022-2023

Le but de ce premier TD est de prendre en main les outils nécessaires à la réalisation de quelques programmes simples, basés sur les exercices réalisés en TD d'algorithmique. Quelques minutes avant la fin du TD, rassembler tous les fichiers C réalisés dans une archive .zip, et la **déposer sur le site du cours** dans l'espace dédié à votre groupe.

---

## Pré-requis

Pour réaliser des programmes en langage C, il faut

- un éditeur de texte pour y taper le code de ces programmes dans des fichiers d'extension `.c`; le code d'un programme étant juste du texte pur, ne surtout pas utiliser un logiciel de traitement de texte; outils suggérés : Notepad++, Sublime Text, ou Emacs
  - le compilateur `gcc` pour détecter d'éventuelles erreurs dans votre code, puis engendrer une application exécutable;  
`gcc` est installé par défaut sous Linux; sous Windows, il faut installer un environnement simulant Linux, Cygwin (<https://cygwin.com/install.html>), et bien sélectionner le package `gcc-core`; sous MacOS X, il faut installer `xcode-select`;
  - une fenêtre de commande dans laquelle exécuter ce compilateur, puis l'application réalisée;
  - créer un dossier dans lequel vos programmes seront rangés.
- 

**Exercice 1 (Premier programme)** *Voici comment créer un premier programme et surtout comment le compiler et l'exécuter.*

- Dans votre éditeur de texte, ouvrir un nouveau fichier appelé `exo1.c`, qu'il faut enregistrer dans le dossier que vous avez créé pour les programmes de ce cours.
- Y taper le code suivant :

```
#include <stdio.h>

int main() {
    printf("Premier affichage");
    return(0);
}
```

- Plus tard, penser à enregistrer régulièrement votre fichier afin de toujours recompiler la dernière version.
- Ouvrir une fenêtre de commande (parfois appelée un terminal), et utiliser la commande `cd` pour vous placer dans le dossier contenant votre fichier `exo1.c` (la commande `pwd` permet d'afficher le dossier dans lequel vous êtes, et la commande `ls` permet de lister le contenu du dossier courant).
- Compiler votre programme grâce à la commande suivante dans votre terminal :  
`gcc exo1.c -o exo1`  
Si vous avez fait des erreurs en recopiant le code, elles vous seront signalées.  
Si votre code est correct, un fichier exécutable `exo1` (ou `exo1.exe` sous Windows) est créé.
- Exécuter votre programme dans votre terminal :  
`./exo1` (sous Linux ou MacOS X) ou `./exo1.exe` (sous Windows)
- Modifier la ligne `printf("Premier affichage");` en `printf("Premier affichage\n");` ; Enregistrer, compiler, exécuter... quelle différence ? (à quoi sert `\n` ?)
- Ajouter des instructions dans votre programme pour que le résultat de l'exécution soit exactement ceci : (utiliser le caractère de tabulation (`\t`) dans le message à afficher)

```
Premier affichage
    un 2e
    et un 3ème !
```

---

**Exercice 2 (Calculs numériques)** Voici un petit exercice pour tester les principaux types numériques proposés en C pour les variables. Créer un nouveau fichier C (*exo2.c*, par exemple) dans un sous-dossier TD1 de votre dossier de travail (comme vous le ferez à chaque nouvel exercice) et y écrire un programme dont la fonction principale (*main*) contient des déclarations de variables (*x1*, *x2*, *x3*, *x4* et *x5* sont des entiers (*int*), *y1*, *y2*, *y3* et *y4* sont des réels en virgule flottante (*float*), et *y5* est un réel au format double (*double*)), puis ajouter les instructions suivantes :

```
x1 = 6;
x2 = x1*2-4;
x3 = x2/3;
x4 = x3-x1;
x5 = x4*x4;
y1 = x2/3;
y2 = x2/3.0;
y3 = (float)x2/3;
y4 = y2*y3;
y5 = y4*y3*y2*y1;
printf("x1=%d x2=%d x3=%d x4=%d x5=%4d\n", x1, x2, x3, x4, x5);
printf("y1=%f y2=%f y3=%4.2f\n", y1, y2, y3);
printf("y4=%5.1f y5=%3.1f\n", y4, y5);
```

Quelles sont les valeurs affichées ? Pourquoi certaines peuvent paraître surprenantes ? (conversion entier/réel, format d'affichage) Assurez-vous de bien comprendre chaque valeur affichée par rapport au calcul correspondant.

---

**Exercice 3 (Calculs de moyennes)** Petits exercices nécessitant l'intervention de l'utilisateur à l'exécution.

**Activité 3.1 (Moyenne)** Écrire un programme qui calcule la moyenne de 3 notes saisies par l'utilisateur à l'exécution (utiliser l'instruction *printf* pour poser une question, et *scanf* pour lire la réponse). Une note est un nombre réel.

**Activité 3.2 (Moyenne pondérée)** Compléter le programme précédent pour calculer la moyenne sachant que les deux premières notes ont un coefficient 1 et la troisième note a un coefficient 2. Écrire une nouvelle version du programme précédent dans lequel le coefficient de chaque note (un entier) est demandé à l'utilisateur.

---

**Exercice 4 (Salaire)** Écrire le programme qui demande à l'utilisateur le montant de ses impôts et son taux d'imposition, puis calcule et affiche le montant de son salaire.

---

**Exercice 5 (Monnaie minimale)** L'objectif est d'écrire un programme qui affiche le nombre minimal de billets de 10€ et 5€, et de pièces de 2€ et 1€, nécessaires pour totaliser un montant *m* donné.

**Activité 5.1 (Version figée)** Programmez cet algorithme, en fixant la valeur de *m* au début du programme (utiliser l'opérateur % calculant le reste de la division entière, aussi appelé modulo).

**Activité 5.2 (Version personnalisée)** Modifier le programme précédent pour que la valeur de *m* soit saisie par l'utilisateur à l'exécution.

---

**Exercice 6 (Carrelages)** Écrire le programme qui permet de calculer le nombre de carrelages carrés de côté 10 cm à acheter pour recouvrir le sol d'une pièce rectangulaire dont les dimensions sont données par l'utilisateur.

Tout carrelage partiellement utilisé n'est plus réutilisable.