



# Initiation à la Programmation C

## Travaux Pratiques - L2

### Variables, instructions, conditionnelles



---

Dans cette séance de travaux pratique, nous abordons les points suivants :

- la compilation d'un programme écrit en C pour obtenir un exécutable ;
- l'écriture de quelques programmes utilisant les formats d'affichage ;
- les types de base ;
- les structures conditionnelles ;
- les boucles.

---

#### ► Exercice 1. Compilation, création dun exécutable

1. Dans votre répertoire personnel, créer un répertoire `Prog_C` ainsi qu'un sous-répertoire `TP_1` soit à l'aide du terminal pour les plus joueurs, soit avec la souris pour les moins joueurs :-).
2. A l'aide de votre éditeur "préféré", créez un fichier dont le nom est `Exo_1_Compilation.c` et écrivez un programme C qui affiche votre nom et votre prénom dans cet ordre, séparé par un espace. Vous ajouterez aussi un retour à la ligne.
3. Compiler ce programme à l'aide du compilateur `clang`, avec l'option `-std=c17`.

Quel est le résultat de la compilation ?

On notera qu'aucun paramètre autre que le nom du fichier à compiler n'a été fourni à l'utilitaire de compilation.

4. Quels sont les droits associés au fichier exécutable créé ? Pourquoi ?
5. Vérifier que l'exécution est correcte.
6. Comment donner le nom CV au fichier exécutable.

#### ► Exercice 2. Messages d'erreur, erreurs de conception

Récupérer le fichier `moyenne.c` sur le `elearning` du cours.

1. A l'aide d'un éditeur, indentez correctement ce programme.
2. Compiler avec `clang` pour obtenir une exécutable de nom `moyenne`
3. Tester et corriger les erreurs de compilation.
4. Tester et corriger les erreurs de conception.

#### ► Exercice 3. Nombre entier vs nombre décimal

Ecrire un programme qui demande à l'utilisateur un nombre et affiche si celui-ci est un nombre entier ou un nombre à virgules.

*Indication* : Le test d'égalité `1 == 1.0` est vrai.

#### ► Exercice 4. FizzBuzz

Ecrire un programme C qui :

1. lit un nombre entier  $n$  strictement positif saisi au clavier
2. affiche tous les entiers de 1 à  $n$ , un par ligne, mais ...
  - affiche le texte "Fizz" à la place de l'entier  $k$  si celui-ci est divisible par 2 ;
  - affiche le texte "Buzz" à la place de l'entier  $k$  si celui-ci est divisible par 3 ;
  - affiche le texte "FizzBuzz" à la place de l'entier  $k$  si celui-ci est divisible par 2 et par 3 ;

#### ► Exercice 5. Mise sous la forme siècles, années, mois, jours d'une durée

Ecrire un programme transformant un nombre de jours saisi par l'utilisateur en nombre de siècles, de mois, de semaines et de jours restant.

Pour simplifier le problème, on considèrera que tous les mois ont 30 jours et toutes les années 360 jours.

Un exemple de sortie du programme pourra être :

```
Saisissez un nombres de jours , svp : -42
Saisissez un nombres de jours , svp : 4242
4242 jours correspondent a :
0 siecle 11 annees 9 mois 1 semaine 5 jours
```

On remarquera l'application des règles d'accord du pluriel : 11 **annees** et non pas 11 **annee**

#### ► Exercice 6. (*Bonus*) Equation de droites dans le plan

1. Décrire l'algorithme permettant de déterminer les coefficients  $a$  et  $b$  de l'équation  $y = ax + b$  d'une droite du plan à partir de deux points n'ayant pas la même abscisse.
2. Ecrire un programme qui affiche l'équation de la droite passant par deux points  $P$  et  $Q$  dont les coordonnées auront été saisies au clavier.

On notera la tentative pour avoir une écriture d'équation de droite la plus proche de celle qu'on utiliserait en mathématiques (pas de  $y = -1.000000\ x + -3.000000$  par exemple...)

Voici quelques exemples de sortie du programme :

Coordonnees du 1ier point : 3 3 Coordonnees du 2nd point : 3 6 Les deux points ont la meme abscisse...	Coordonnees du 1ier point : 1 6 Coordonnees du 2nd point : -1 1 Equation de droite : $y = 2.500000\ x + 3.500000$
Coordonnees du 1ier point : 2 -5 Coordonnees du 2nd point : -4 1 Equation de droite : $y = -x - 3$	Coordonnees du 1ier point : 2 6.5 Coordonnees du 2nd point : 1 4.5 Equation de droite : $y = 2\ x + 2.500000$