

# TD1 : Introduction au langage C (Partie2)

IN323 - Programmation en langage C++

Elisa HENRION-GUENEAU

IPSA - Site Toulouse

AERO3 – Année 2025/2026

*Consigne générale : En C, la gestion de la mémoire est explicite. Soyez vigilants sur l'utilisation des adresses (&) et des pointeurs (\*). N'oubliez jamais de libérer la mémoire allouée dynamiquement.*

## Exercice 4 : Pointeurs

1. Déclarer une variable `cap` (entier) puis créer un pointeur `ptr_cap` pointant sur cette variable.  
Modifier la valeur du `cap` uniquement via le pointeur pour simuler une correction de trajectoire.  
Afficher la valeur avant et après modification.
2. Lors d'un vol stationnaire, un drone doit calculer sa consommation.  
Écrire une fonction `void energie(float tension, float courant, float *puissance, float *resistance)` qui calcule la puissance ( $P = U \times I$ ) et la résistance ( $R = U/I$ ) et "renvoie" ces deux valeurs via des pointeurs.

## Exercice 5 : Tableaux et Allocation Dynamique

Un avion de test enregistre des pressions hydrauliques à intervalles réguliers.

- Écrire un programme qui demande à l'utilisateur le nombre de mesures  $N$  effectuées pendant le vol.
- Allouer dynamiquement un tableau de flottants de taille  $N$  pour stocker ces mesures.
- Remplir le tableau avec les valeurs saisies par l'utilisateur
- Écrire une fonction pour calculer la pression moyenne du vol.
- Afficher la moyenne et libérer la mémoire.

## Exercice 6 : Chaînes de caractères

1. Écrire une fonction `len` qui calcule la longueur d'un message radio (chaîne de caractères) sans utiliser la bibliothèque `string.h`.
2. Écrire une fonction `cpy` qui recopie le contenu d'un enregistreur de vol (chaîne de caractères) vers un serveur de secours (chaîne de caractères), sans utiliser la bibliothèque `string.h`.
3. Dans certains protocoles, un code de sécurité doit être un "palindrome" pour être validé.  
Écrire la fonction `palindrome` qui vérifie si une séquence de caractères (ex : "RADAR") est identique dans les deux sens.

## Exercice 7 : Structures

1. Définir une structure `Avion` contenant un identifiant (chaîne), une altitude (entier) et une vitesse (float).
2. Écrire une fonction `montee(Avion *a, int gain)` qui prend un pointeur vers un avion et augmente son altitude du gain spécifié.
3. Dans le `main`, créer un avion "Airbus-A320", initialiser ses données, et utiliser la fonction `montee` pour passer de 2000m à 3000m. Afficher le résultat en utilisant l'opérateur `->`.