

TP Langage C IENAC19

Modalités de passage du TP noté :

L'énoncé du tp se trouve sur ce document. **Suivez bien les instructions suivantes :**

- Vous allez vous connecter sous Ubuntu en utilisant le login « enacaudio », mot de passe « enacaudio » (et **non pas votre login élève !**).
- Créez sur le bureau un répertoire dont le nom sera composé de votre nom, votre prénom ainsi que la promo (exemple « DUPONT_Alain_IENAC19 »).
- Vous travaillerez dans ce répertoire et laisserez tous vos fichiers créés durant le TP noté.
- Vous allez trouver sur le Bureau un fichier zip contenant les fichiers et le sujet du tp noté.
- **Supprimez tous les autres répertoires et documents qui se trouvent sur le bureau.**
- A la fin de l'épreuve, vous ne laisserez sur le bureau que votre répertoire de travail, vous supprimerez les autres documents.
- N'oubliez pas de laisser uniquement votre répertoire de travail sur le bureau. **Vous laisserez ensuite la session ouverte sans éteindre le PC et quitterez la salle.**

Sujet :

Nous allons travailler avec des données sur des avions : pour chaque avion, nous aurons son indicatif, sa marque, son type, son nombre de places et son nombre de places réservées.

Vous créerez pour cela un projet avec les fichiers d'en-tête et les fichiers source C correspondant. Un *main* vous est donné avec ce sujet.

Vous répondrez à la partie 1 en premier et ensuite à la partie 2.

Vous testerez les fonctions demandées avec le *main* fourni (et le fichier texte « avions.txt »).

*

Questions Partie 1

Question 1 (1 pt):

Définissez en Langage C un nouveau type utilisateur `Avion`. Ce type sera une structure qui possédera 5 champs : `indicatif` (pour l'indicatif de l'avion de type chaîne de caractères de longueur 20), `marque` pour la marque de l'avion de type chaîne de caractères de longueur 20, `type` pour le type de l'avion de type chaîne de caractères de longueur 20, `nbplaces` pour le nombre de places de l'avion de type `int`, `nbresa` pour le nombre de places réservées de type `int`.

Question 2 (3 pts):

Définissez la fonction « `void saisir(int * nb, Avion *t)` » qui saisit au clavier **les données d'un avion** et les stocke à la fin du tableau (dans la première structure de libre, c'est-à-dire celle à l'indice donné par `nb`). On vérifiera que la capacité maximale du tableau ne soit pas dépassée.

Question 3 (3 pts):

Définissez la fonction « `void afficher(int nb, Avion *t)` » qui permet d'afficher à l'écran les caractéristiques de chaque avion du tableau (second paramètre).

Question 4 (3 pts):

Définissez la fonction « `void calculerOccupation(int nb, Avion * t)` » qui affiche à l'écran le pourcentage d'occupation de chaque avion, c'est-à-dire le rapport entre le nombre de places réservées et le nombre de places de l'avion. Par exemple pour un avion d'indicatif « FGCDE », on obtiendra un affichage comme celui-ci :

FGCDE taux occupation = 60.00%

Question 5 (4 pts):

Définissez la fonction « `void charger(int *nb, Avion *t, char * nomfichier)` » qui permet de récupérer dans le fichier texte dont le nom est passé en troisième paramètre, les caractéristiques d'avions (un par ligne) pour remplir les champs des structures d'un tableau de structures Avion. Le premier paramètre de la fonction permet de mettre à jour le nombre total de structures, le second paramètre d'accéder au tableau de structures.

Vous utiliserez le fichier « avions.txt » pour tester votre fonction.

Question 6 (2 pts):

Définissez la fonction « `void sauver(int nb, Avion *t, char * nomfichier)` » qui permet de sauvegarder dans le fichier texte dont le nom est passé en paramètre, les données des avions comme dans le fichier « avions.txt ».

*

Questions Partie 2 : tableau dynamique d'avions

Question 7 (3 pts):

Définissez la fonction « `void saisirDyn(int * nb, Avion **t)` » qui saisit au clavier **les données d'un avion** et les stocke à la fin du tableau dynamique : le tableau sera pour cela agrandi dynamiquement et le nouvel avion ajouté à la fin.

Question 8 (1 pt):

Définissez la fonction « `void liberer(int *nb, Avion **t)` » qui permet de libérer en mémoire les structures précédemment allouées et mettre à jour les paramètres.