

TP Langage C IENAC17

Modalités de passage du TP noté :

L'énoncé du tp se trouve sur ce document (partie1 et 2). Il est conseillé de faire les parties dans l'ordre. **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_IENAC17_AVI »).
- Lancez ensuite votre navigateur, connectez-vous dans la boîte de connexion avec **votre login élève et votre mot de passe**. Allez sur ECampus Enac , dans votre cours de Langage C, dernière section (« TP noté IENAC17 »). Téléchargez les trois fichiers « main1.c », « main2.c » et « montrain.txt » et enregistrez les dans le répertoire précédent du bureau.
- Vous travaillerez dans ce répertoire et laisserez tous vos fichiers créés durant le TP.
- A la fin de l'épreuve vous déposerez sous ECampus Enac dans la dernière section tous les fichiers contenant votre code (compressez votre dossier et déposez-le dans ECampus).
- N'oubliez pas de laisser le répertoire sur le bureau, sans rien effacer. Vous fermerez ensuite la session sans éteindre le PC et quitterez la salle.

Sujet :

Nous allons travailler avec des données récupérées dans un fichier texte qui décrivent un train. Ce dernier est constitué d'un ensemble de wagons. Un wagon possède les caractéristiques suivantes :

-son type : une chaîne de caractères (pouvant avoir les valeurs suivantes : LOCOMOTIVE, VOITURE, MARCHANDISE).

-son identifiant (un entier).

-sa data : un wagon LOCOMOTIVE possède une valeur 0 (zéro), un wagon VOITURE possède le nombre de passagers admissibles, un wagon MARCHANDISE possède la charge utile de transport.

Vous créerez pour cela un projet avec les fichiers d'en-tête et les fichiers source C correspondant.

*

Partie 1 : tableau de structures

Vous testerez les fonctions suivantes avec le main1 fourni (et le fichier texte « montrain.txt »).

Question 1 (1 pt):

Définissez en Langage C un nouveau type utilisateur wagon. Ce type sera une structure qui possédera 3 champs : type (pour le type du wagon, de type chaîne de caractères), ident (pour l'identifiant du wagon, de type int) et data (pour la valeur de sa data, de type int).

Question 2 (2 pts):

Définissez la fonction « void afficherTrain(int index, wagon * train) » qui permet d'afficher à l'écran les caractéristiques de chaque wagon du tableau de wagons « train ». Pour une LOCOMOTIVE, on affichera uniquement son type et son ident, pour une VOITURE son type, son ident et son nombre de passagers, pour un MARCHANDISE son type, son ident et sa charge.

Question 3 (2 pts):

Définissez la fonction « int calculNbPaxTrain(int index, wagon *train) » qui retourne le calcul du nombre total de passagers transportés par le train (on fera la somme des passagers des wagons de type VOITURE).

Question 4 (2 pts):

Définissez la fonction « `int calculChargeTotaleTrain(int index, wagon *train)` » qui retourne le calcul de la charge totale de marchandises transportées par le train (on fera la somme des charges des wagons de type MARCHANDISE).

Question 5 (2 pts) :

Définissez la fonction « `void ajouterWagon(char *type, int ident, int charge, int *index, wagon *train)` » qui permet d'ajouter à la fin du train le nouveau wagon dont les caractéristiques sont passées en paramètre.

Question 6 (4 pts):

Définissez la fonction « `void chargerTrain(char * nomfichier, int *index, wagon * train)` » qui permet de récupérer dans le fichier texte dont le nom est passé en paramètre, les caractéristiques de chaque wagon (un par ligne) pour remplir un tableau statique de wagons. Le paramètre index permet de récupérer la taille de ce tableau.

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

Question 7 (4 pts):

Définissez la fonction « `void sauverTrain(char * nomfichier, int index, wagon * train)` » qui permet de sauvegarder dans le fichier texte dont le nom est passé en paramètre, les caractéristiques de chaque wagon (un par ligne, comme dans le fichier « montrain.txt »).

*

Partie 2 : tableau dynamique de structures

Vous testerez les fonctions demandées ci-dessous avec le main2 fourni. Vous pouvez également utiliser certaines fonctions de la partie 1 compatibles.

Question 1 (1 pt):

Définissez la fonction « `void chargerTrainDyn(char * nomfic, int *index, wagon ** train)` » qui permet de récupérer dans le fichier texte dont le nom est passé en paramètre, les caractéristiques de chaque wagon (un par ligne) pour remplir un tableau dynamique (en mémoire dynamique) de wagons. Le paramètre index permet de récupérer la taille de ce tableau.

Le tableau à construire sera un tableau dynamique de structures et non pas un tableau de pointeurs sur structures.

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

Question 2 (1 pt):

Définissez la fonction « `void ajouterWagonDyn(char *type, int ident, int charge ,int *index, wagon **train)` » qui permet d'ajouter à la fin du train le nouveau wagon dont les caractéristiques sont passées en paramètre. On pourra utiliser la fonction « `realloc` » pour agrandir le tableau dynamique existant.

Question 3 (1 pt):

Définissez la fonction « `void libererTrain(int *index, wagon **train)` » qui permet de libérer en mémoire le tableau précédemment alloué et mettre à jour les paramètres.