Système d'exploitation

TP 1 - Rappels / compléments de programmation C

Les premières séances de TP du module SE sont dédiées à la révision / compléments des notions principales de programmation C telles que : fonction, structure, allocation dynamique, écriture et lecture dans les fichiers ainsi que la compilation séparée.

L'objectif du TP est d'écrire un programme permettant de récupérer les coefficients de deux matrices dont les coefficients sont enregistrés dans des fichiers différents, et faire une opération (uniquement la multiplication matricielle) sur ces deux matrices, puis d'enregistrer le résultat dans un fichier.

L'ensemble des fichiers sources seront séparés en plusieurs modules : un module de lecture/écriture dans des fichiers, un module de gestion de matrices qui s'appuiera sur un module de gestion de tableau 2D. Chaque module sera codé et testé de manière indépendante.

Note: Vous aurez besoin des fonctions suivantes tout au long du TP : fopen, fclose, fprintf, sscanf, fputs, fgets, strcpy, strcomp, strlen, malloc, realloc, free.

1 Module de gestion d'un tableau 2D de réels

Cette partie concerne l'écriture d'un ensemble de fichiers (.c et .h) permettant de gérer un tableau 2D de réels. Vous devrez écrire un fichier permettant de tester le module (i.e. l'ensemble des fonctions).

Question 1. Ce module doit contenir les fonctions suivantes :

- Une fonction chargée d'allouer un tableau 2D de réels dynamiquement. Les nombres de lignes et de colonnes du tableau sont connus lors de l'appel. **Deux versions sont à faire : une version "non void"** qui retourne le tableau créé et **une version "void"** qui ne retourne rien. Commencer par coder la fonction "non void" qui est la plus simple;
- Une fonction chargée de saisir l'ensemble des réels du tableau 2D;
- Une fonction permettant d'afficher le contenu du tableau 2D dans le terminal (sous la forme d'un tableau);
- Une fonction chargée de désallouer un tableau alloué dynamiquement par la fonction d'allocation.

2 Module de gestion d'une matrice

Cette partie concerne l'écriture d'un ensemble de fichiers (.c et .h) permettant de gérer des matrices. **Ce module doit s'appuyer sur le module de gestion de tableau 2D écrit précédemment**. Vous devez donc réutiliser au maximum les fonctions du module précédemment.

Question 2. Ce module doit contenir :

- La définition d'un type *t_matrice* contenant toutes les informations d'une matrice;
- Une fonction permettant d'allouer dynamiquement une matrice;
- Une fonction permettant de saisir les coefficients d'une matrice à l'aide de saisies utilisateurs (La récupération des coefficients d'une matrice dans un fichier sera ajouté par la suite au module);
- Une fonction permettant d'afficher une matrice;
- Une fonction permettant d'effectuer le produit matriciel;
- Une fonction permettant de désallouer une matrice.

3 Module de gestion d'écriture/lecture dans un fichier

Cette partie concerne l'écriture d'un ensemble de fichiers (.c et .h) permettant d'effectuer des écritures et lectures dans des fichiers.

Question 3. Écrire les fonctions suivantes :

- Une fonction permettant de lire tout le contenu d'un fichier. La fonction a comme entrée le nom du fichier à lire et retournera sous la forme d'une chaîne de caractères (dynamique) le contenu du fichier lu;
- Une fonction permettant d'écrire une chaîne de caractère dans un fichier dont le nom est donné en entrée de la fonction. Si le fichier existe tout le contenu sera écrasé;
- Une fonction permettant de copier tout le contenu d'un fichier en s'appuyant sur les deux fonctions précédentes.

Question 4. Ajouter la possibilité dans le module de gestion des matrices de pouvoir récupérer les coefficients dans un fichier. Vous devez pour cela utiliser la fonction de lecture écrite précédemment. Le problème étant que la fonction renvoie tout le contenu du fichier sous la forme d'une chaine de caractères. Vous devrez donc écrire une fonction permettant d'extraire chaque coefficient de cette chaine de caractère.

Le format du fichier contenant les coefficients d'une matrice est de la forme

- La 1ère ligne contiendra 2 nombres représentant les dimensions de la matrice (nombre de lignes et nombre de colonnes);
- Les lignes suivantes contiendront les coefficients de la matrice;

Par exemple, le contenu du fichier d'une matrice ayant 2 lignes et 3 colonnes dont les coefficients de la 1ère ligne (de la matrice) sont [4 5 6] et les coefficients de la 2ème ligne (de la matrice) sont [7 8 9] est :

```
2 3
4 5 6
7 8 9
```

Note: Nous aurions pu réécrire simplement la fonction de lecture de fichier pour l'adapter à la lecture d'un fichier formaté pour une matrice mais l'idée ici est justement de ne pas réécrire la fonction mais d'utiliser la fonction précédemment codée (comme si elle avait été écrite dans une bibliothèque par un autre programmeur).

Vous devrez donc lire "mot par mot" la chaine de caractères renvoyée par la fonction de lecture. Nous utiliserons la fonction *sscanf()* pour faire cela. Cependant cette fonction ne fonctionne pas comme les fonctions fread/fwrite ou une variable de position est mise à jour automatiquement. Ce sera à vous de gérer cette variable de position sinon le *sscanf()* reliera toujours le premier mot de la chaîne de caractère... Lorsque vous envoyez une chaine de caractères à une fonction c'est l'adresse du début de la chaîne que vous envoyez, rien ne vous empêche d'effectuer un décalage sur cette adresse au moment de l'appel.

La fonction à écrire peut être appelée *atomatrice()* (ASCII to matrice). On peut noter la similitude avec les fonctions telles que atoi ou atof disponibles dans les bibliothèques standards.

4 Options au lancement du programme

Question 5. Votre programme (final) doit permettre à l'utilisateur de donner les informations de noms de fichiers à lire (2 fichiers dans notre cas) ou à produire en sortie (1 seul fichier dans notre cas : produit matriciel).

Lors du lancement si le nombre de paramètres sont erronés la commande doit retourner à l'utilisateur la possibilité d'utiliser le flag *–help* pour connaître le format de lancement. Exemple format d'affichage de l'aide après un lancement incorrect (il manque un nom ou les noms des fichiers de matrices n'existent pas) :

```
./a.out
  Utiliser le flag --help pour consulter l'aide
```

Seul les noms des fichiers d'entrée (ceux ou extraire les coefficients des matrices) sont obligatoires. Le nom du fichier de sortie (contenant le produit matriciel) peut être indiquer au programme après le flag *-o* (output). Le nom par défaut de ce fichier de sortie est *out.txt*.

On prendra pour exemple le format d'affichage du help de gcc :

```
./a.out --help
   Utilisation: a.out [option] fichier1 fichier2
   Options :
     --help    Afficher ces informations
     -o <s>    Produit matriciel dans le fichier de sortie <s>, par defaut <out.txt>
```

Exemples corrects de lancement du programme :

```
./a.out matrice1.txt matrice2.txt -o result.txt
   --> resultat du produit matriciel dans result.txt
./a.out -o sortie.txt m1.txt m2.txt
   --> resultat du produit matriciel dans sortie.txt
./a.out matrice1.txt matrice2.txt
   --> resultat du produit matriciel dans out.txt
```