Examen de rattrapage - ING1 GI-GM

Programmation C

21 Février 2019

Modalités

- Durée : 3 heures
- Toutes vos affaires (sacs, vestes, trousse, etc.) doivent être placées à l'avant de la salle.
- Aucun document papier n'est autorisé.
- Aucune sortie n'est autorisée avant une durée incompressible de 1 heure.
- Aucun déplacement n'est autorisé.
- **Aucune question au professeur n'est autorisée.** Si le sujet ne vous semble pas clair, à vous d'expliquer dans des commentaires pourquoi est-ce que ça ne vous semble pas clair, et quelles modifications vous effectuez pour réaliser l'exercice.
- Aucun échange, de quelle que nature que ce soit, n'est possible.
- Les différentes fonctions/procédures demandées seront toutes à tester dans un **main**.
- Vous traiterez chaque exercice dans un dossier différent.
- Le barème est indicatif, il intègre pour chaque question un test dans un main.
- La présence d'un Makefile et des commentaires doxygens sont un plus, et seront appréciés.
- La lisibilité du code et les commentaires seront pris en compte dans la notation, de même que la séparation des différentes procédures/fonctions dans des fichiers d'en-tête (.c .h).
- **Tout code qui ne compile pas ne sera pas corrigé et entraînera une pénalité de 50%. Si une partie de votre code ne compile pas, commentez-le. Ceci implique évidemment que vous compiliez au fur et à mesure!**
- Chaque **warning** de compilation entraînera une **pénalité de 15%.**
- Tout ce qui sera dans le dossier rendu correspondra à votre rendu, et sera récupéré à la fin de l'examen. Je vous conseille vivement de travailler directement dans ce dossier.

Exercice 1 5 points

Écrire une fonction *premiereOccurrence* qui prend en paramètre deux chaînes de caractères *str1* et *str2*. Elle retourne l'adresse dans la chaîne *str1* de la première occurrence de la chaîne *str2*. Si *Str2* n'existe pas dans *Str1*, la fonction retourne *null*.

Ecrire un programme qui teste cette fonction.

NB: Développez votre propre fonction, l'usage d'une fonction de C qui réalise ce genre de traitement (comme par exemple *strstr*) est interdit.

Exercice 2 8 points

Supposant un fichier *commandes.txt* contenant une suite de lignes décrivant chacune un commande reçue.

Chaque ligne du fichier contient 4 types d'information séparés par un espace: le nom du client, le nom du produit, la quantité commandée du produit et son prix unitaire.

Exemple de deux lignes extraites du fichier:

Paul ordinateur 5 400
Marie classeur 200 2

a) Définir une structure *Commande* pour contenir les 4 types d'information contenus dans une ligne : *nomC* (nom du client), *nomP* (nom du produit), *qteCom* (quantité commandée), *prixP* (prix du produit).

b) Ecrire une fonction qui lit le contenu du fichier *commandes.txt* et remplit un tableau *commandesTab* de type *Commande*. La taille du tableau est égale au nombre de lignes dans le fichier. La fonction retourne le tableau rempli.

c) Ecrire une fonction qui affiche le contenu du tableau *commandesTab*.

d) Ecrire une fonction qui retourne la moyenne des valeurs des commandes, sachant que la valeur d'une commande = *qteCom* * *prixP*.

e) Ecrire une procédure qui écrit dans un fichier *commandesInf.txt*, toute commande dont la valeur (*qteCom* * *prixP*) est inférieure à la moyenne.

f) Ecrire une fonction qui retourne l'indice dans le tableau *commandesTab* de la commande ayant la valeur (*qteCom* * *prixP*) la plus proche de la moyenne.

Exercice 3 7 points

Dans cet exercice nous recherchons les points-clos d'une matrice à deux dimensions. Un point-clos est un élément d'une matrice qui est à la fois: un maximum sur la ligne et un minimum sur la colonne.

Exemples:

5	2	4		3	**8**
6	3	9		2	10
8	6	5		1	**8**

Dans la première matrice, 5 est le maximum sur la ligne et le minimum sur la colonne. Dans la seconde matrice, 8 est le maximum sur la ligne et le minimum sur la colonne.

a) Ecrire une fonction *saisirMatrice* qui demande à l'utilisateur le nombre de lignes et le nombre de colonnes d'une matrice *A*, ainsi que ses éléments. La fonction retourne la matrice saisie.

b) Ecrire une fonction *afficherMatrice* qui affiche les éléments d'une matrice passée en paramètre.

c) Ecrire une fonction *lesMaxDesLignes* qui prend en paramètre une matrice *A* et crée une nouvelle matrice *MaxDesLignes* de mêmes dimensions, tel que:
   - *MaxDesLignes[i,j]* = 1 si *A[i,j]* est un maximum sur la ligne i de *A* et
   - *MaxDesLignes [i,j]* = 0 sinon.

d) Ecrire une fonction *lesMinDesColonnes* qui prend en paramètre une matrice *A* et crée une nouvelle matrice *MinDesColonnes* de mêmes dimensions, tel que:
   - *MinDesColonnes[i,j]* = 1 si *A[i,j]* est un minimum sur la colonne j de *A* et
   - *MinDesColonnes[i,j]* = 0 sinon.

e) Ecrire une fonction *TrouverPointsClos* qui trouve et affiche les valeurs et indices (i,j) de tous les points-clos d'une matrice *A*.