Université Mohammed V-Agdal
Faculté des Sciences de Rabat
Département d'Informatique

Année Universitaire 2019/2020
MODULE M21 (SMI4)

Travaux Dirigés de Programmation II
[TD n°4 : Les Fichiers en C

Objectifs : *- Manipulation en C, des fichiers (de texte et binaires).*

Exercice 1

On dispose du **fichier** "produits.txt" suivant :

73	Ordinateur	5000.00
07	Cle_8GO	120.00
15	Disque_Externe	960.00
…	…	…
…	…	…

Chaque ligne de ce **fichier de texte** décrit un produit : le code du produit (*un entier*), le nom du produit (*une chaîne de caractères*) et le prix du produit (*un réel*).

a)- Ecrire un programme ***C***, qui permet la lecture, **caractère par caractère**, du contenu du fichier et l'affiche à l'écran.

b)- Refaire a)- à l'aide d'une lecture **ligne par ligne**.

c)- Refaire a)- en utilisant une lecture **formatée**.

Exercice 2

Ecrire une fonction, **`save_tab_reels`**, qui permet de sauvegarder un tableau de réels dans un **fichier de texte** de nom donné.

Exercice 3

a)- Ecrire une fonction, **`compte_car(FILE * f)`**, qui retourne le nombre de caractères présents dans un **fichier de texte**.

b)- Ecrire une fonction, **`compte_mot(FILE * f)`**, qui retourne le nombre de mots présents dans un fichier. Les mots sont séparés par des espaces ou des retours à la ligne.

c)- Ecrire une fonction, **`compte_ligne(FILE *f)`**, qui retourne le nombre de lignes d'un fichier.

Exercice 4

a)- Écrire une fonction, **`foccur_mot`**, qui lit un nom de fichier au clavier, ouvre un fichier de ce nom et compte le nombre d'occurrences d'un mot dans le fichier. On supposera que le mot ne contient pas d'espace et que le fichier ne contient pas de ponctuation. Le mot doit être passé en paramètre.

b)- Écrire un programme ***C*** pour tester la fonction.

Exercice 5 (facultatif)

Écrire une fonction, **`compte_while_ligne`**, qui prend en paramètre le nom d'un fichier texte et détermine le nombre de lignes du fichier qui commencent par le mot "while".

Exercice 6

Ecrire un programme ***C*** qui permet de lire les données du fichier de texte "produits.txt" (*supposé déjà créé, voir exercice 1*) et les stocke dans le **fichier binaire** "produits.bin". Les données sont stockées dans le même ordre du fichier de texte.

Exercice 7

Ecrire la fonction, **`int ma_fgetc(FILE *f)`**, qui agit comme la fonction **`fgetc`**. **Indication** : utiliser la fonction **`fread`**.

Exercice 8

On considère un fichier binaire qui contient un entier et des nombres réels. Le premier élément du fichier (de type int) donne le nombre d'éléments (chacun de type float) qui se trouvent à la suite. Ecrire une fonction, **`float *charger(char *nomfichier, int *n)`**, qui permet de charger les éléments du fichier dans un tableau en mémoire.

Exercice 9

Ecrire une fonction, **`void sauvegarder(float *tab, int n, char *nomfichier)`**, qui permet de sauvegarder dans un fichier binaire, un tableau tab contenant n nombres réels. Le format du fichier est le même que celui de la question précédente : on trouve d'abord le nombre d'éléments, puis les éléments à la suite dans le fichier.

Exercice 10

On considère le fichier binaire "produits.bin" créé par l'exercice **6** ci-dessus.

a)- Ecrire une fonction, **`int recherche_par_nom (FILE *f, char *nomp, produit p`**), qui à partir du fichier de produits, permet de retrouver les informations correspondant à produit de nom donné. La fonction retourne **1** en cas de succès, sinon **0**.

b)- Ecrire une fonction, **`int recherche_par_rang(FILE *f, int rangp, produit p)`**, qui à partir du fichier de produits, permet de retrouver les informations relatives à un produit de rang donné (par accès direct). La fonction retourne **1** en cas de succès, sinon **0**.

c)- Ecrire une fonction, **`void modifie(FILE *f, int codep)`**, qui étant donné le code d'un produit, permet de modifier l'enregistrement correspondant, en augmentant son prix de 10%.

Exercice 11 (facultatif)

On considère deux fichiers formés d'enregistrements comportant un champ, appelé **clé**, qui est une chaîne de caractères. Ces deux fichiers sont triés : leurs articles se suivent dans l'ordre croissant des clés (*il s'agit de l'ordre lexicographique*).

Ecrire un programme ***C***, qui permet de fusionner ces deux fichiers en un seul, également trié.