TP 9 : Fichiers

Gestion d'un stock de magasin, suite et fin.

Partie 1 :

En reprenant le programme de gestion de stocks d'un magasin (TP8 partie 1 ou 2 suivant le cas), ajoutez la fonctionnalité qui permet d'initialiser votre tableau stock en début de session (au début de la fonction main) à partir d'un fichier texte qui aura été préalablement créé sous un éditeur de texte.
Pour constituer ce fichier texte, vous respecterez la convention suivante :

- le fichier texte se trouvera dans le même répertoire que le programme,
- le fichier texte sera structuré avec un article par ligne
- la référence, le prix et la quantité seront séparés par un caractère blanc
- la désignation sera la chaîne de caractères qui suit la quantité et qui se termine par la fin de ligne

Exemple de fichier texte représentant un stock :

```
3681  1.10  21  aspirateur
5234  2.3  35  lave-linge
8167 15.0  18 télévision
581  3.20  500 four
```

Partie 2 :

Le fichier texte a permis d'initialiser votre stock mais n'est pas la meilleure manière de mémoriser l'information et on préférera utiliser ensuite un fichier binaire. Pour cela, vous allez ajouter les fonctionnalités suivantes dans votre programme :

- Avant de quitter l'application, créez dans le répertoire courant un **fichier binaire** "stock.dat" contenant la copie des articles stockés en mémoire centrale.
- En début de session (au lancement de votre programme), si le fichier binaire "stock.dat" existe déjà, lire les données du stock dans ce fichier. Sinon (erreur détectée à l'ouverture), lire les données dans le fichier texte "stock.txt" comme précédemment.

Quelques rappels de cours

ouverture et fermeture d'un flot:

```
FILE *fopen(const char *nom, const char *mode)
```

"r" (read) ouverture en lecture d'un fichier existant
"r+" idem "r" mais écriture permise
"w" (write) ouverture en écriture d'un fichier existant ou non (écrasé s'il existe)
"w+" idem "w" mais lecture permise (différent de "r+")
"a" (apend) allongement d'un fichier. Le système se positionne en fin de fichier et seules les écritures sont permises
"a+" idem "a" mais lectures permises

```
int fclose(FILE *flot)
```

lecture, écriture dans un fichier binaire et positionnement :

```
int fread(void *adr, int taille, int nombre, FILE *f )
int fwrite(const void *adr, int taille, int nombre, FILE *f )

int rewind (FILE *f)
int fseek(FILE *f, long saut, int origine )
```

en prenant pour origine SEEK_CUR (position courante), SEEK_END (fin), SEEK_SET (début).

Les fichiers textes : principales fonctions

`char fgetc(FILE *f)`
rend le caractère suivant dans le flot (EOF si fin de fichier)

`char fputc(char caractere, FILE *flot)`
écrit le caractère dans le flot, rend le caractère ou EOF si une erreur se produit.

`char *fgets(char *s, int n, FILE *flot)`
lit une ligne (terminée par '\n') d'au plus (n-1) caractères et la recopie dans le tableau pointé par s. Un caractère '\0' est ajouté en fin (à la suite de \n) pour respecter la structure d'une chaîne. La fonction rend s ou NULL.

`int fputs(const char *s, FILE *flot)`
écrit dans le flot la chaîne pointée par s.

`int fscanf(FILE *f, format, adr1, adr2, ..., adrn)`
lit des caractères dans le flot, les convertit en n valeurs et les affecte aux adresses indiquées. Cette fonction retourne le nombre de variables effectivement lues ou EOF si la fin de fichier est atteinte ou si aucune lecture n'est faite.

`int fprintf(FILE *f, format, expr1, expr2, ..., exprn)`
écrit dans le flot les résultats des n expressions.