

## Programmation II : SMI (S4)

### TP N° 4 : Les Fichiers (Correction)

#### Partie I : Exercices préliminaires

##### Exercice 1 :

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

struct Etudiant{
    char nom[30];
    char prenom[30];
    char sem[3];
    float moy;
};

typedef struct Etudiant Etudiant;

void recupEtudiants(char nomFich[100], Etudiant ** tabE, int * nbE){
    FILE * pf;
    pf = fopen(nomFich, "r");
    if(pf == NULL){
        printf("Impossible d'ouvrir le fichier !!!");
        exit(1);
    }
    else{
        int i;
        fscanf(pf, "%d", nbE);
        (*tabE) = (Etudiant*)malloc((*nbE) * sizeof(Etudiant));
        for(i=0;i<(*nbE);i++){
            fscanf(pf, "%s %s %s %f", (*tabE)[i].nom,
                (*tabE)[i].prenom, (*tabE)[i].sem, &(*tabE)[i].moy);
        }
    }
    fclose(pf);
}

void afficherEtudiants(Etudiant * tabE, int nbE){
    int i;
    for(i=0;i<nbE;i++){
        printf("Etudiant : %s %s %s %.2f\n", tabE[i].nom,
            tabE[i].prenom, tabE[i].sem, tabE[i].moy);
    }
}

int main(){
    Etudiant * TE;
    int taille;
    recupEtudiants("etudiants.txt", &TE, &taille);
    afficherEtudiants(TE, taille);

    return 0;
}
```

## Partie II : Problème (Gestion d'une pharmacie)

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

// Qestion 1
struct Client{
    char nom[20];
    double credit;
};
typedef struct Client Client;

struct Medicament{
    char libelle[20];
    double prix;
    int qteStock;
};
typedef struct Medicament Medicament;

typedef struct Pharmacie{
    Client * tabC;
    Medicament * tabM;
    int nbC;
    int nbM;
    int MAXC;
    int MAXM;
    double solde;
};

void initClient(Client * c, char n[], double crdt){
    strcpy(c->nom, n);
    c->credit = crdt;
}

void afficherClient(Client c){
    printf("Client : %s %.2f\n", c.nom, c.credit);
}

void initMedicament(Medicament * m, char lib[], double p, unsigned q){
    strcpy(m->libelle, lib);
    m->prix = p;
    m->qteStock = q;
}

void afficherMedicament(Medicament m){
    printf("Medicament : %s %.2f %d\n", m.libelle, m.prix, m.qteStock);
}

void initPharmacie(Pharmacie * adrPh, double s, int maxC, int maxM){
    adrPh->solde = s;
    adrPh->nbC = 0;
    adrPh->nbM = 0;
    adrPh->MAXC = maxC;
    adrPh->MAXM = maxM;
    adrPh->tabC = (Client*)malloc(adrPh->MAXC * sizeof(Client));
    adrPh->tabM = (Medicament*)malloc(adrPh->MAXM * sizeof(Medicament));
}

void enregistrerClient(Pharmacie * adrPh, Client c){
    if(adrPh->nbC < adrPh->MAXC){
```

```

        adrPh->tabC[adrPh->nbC++] = c;
    }
    else{
        printf("Impossible d'ajouter un client !!!\n");
    }
}

void enregisterMedicament(Pharmacie * adrPh, Medicament m){
    if(adrPh->nbM < adrPh->MAXM){
        adrPh->tabM[adrPh->nbM++] = m;
    }
    else{
        printf("Impossible d'ajouter un médicament !!!\n");
    }
}

int verifMedicament(Pharmacie ph, char nMed[]){
    int i;
    for(i=0; i<ph.nbM; i++){
        if(strcmp(ph.tabM[i].libelle, nMed)==0){
            return i;
        }
    }
    return -1;
}

void approvisionner(Pharmacie * adrPh, char nMed[], int q){
    int pos = verifMedicament(*adrPh, nMed);
    if(pos!=-1){
        if(adrPh->solde >= adrPh->tabM[pos].prix * q){
            adrPh->tabM[pos].qteStock += q;
            adrPh->solde -= adrPh->tabM[pos].prix * q;
        }
        else{
            printf("Pas de crédit !!!\n");
        }
    }
    else{
        printf("Médicament inexistant !!!\n");
    }
}

int verifAchat(Pharmacie ph, char nClient[], char nMed[], int * posClient, int *
posMed){
    (*posClient) = -1;
    // Recherche de la position du client c1 s'il existe
    for(int i = 0; i<ph.nbC; i++){
        if(strcmp(ph.tabC[i].nom, nClient)==0){
            (*posClient) = i;
            break;
        }
    }
    // Recherche de la position du médicament m2 s'il existe
    (*posMed) = -1;
    for(int i = 0; i<ph.nbM; i++){
        if(strcmp(ph.tabM[i].libelle, nMed)==0){
            (*posMed) = i;
            break;
        }
    }
}

void achat(Pharmacie * adrPh, char nClient[], char nMed[], int q){
    int posClient, posMed;

```

```

    verifAchat(*adrPh, nClient, nMed, &posClient, &posMed);
    if(posClient != -1 && posMed != -1){
        if(adrPh->tabM[posMed].qteStock >= 1){
            adrPh->tabM[posMed].qteStock --;
            adrPh->solde += adrPh->tabM[posMed].prix * 1;    // Quantité = 1
            adrPh->tabC[posClient].credit -= adrPh->tabM[posMed].prix * 1;
        }
        else{
            printf("La quantité demandée est non disponible !!!\n");
        }
    }
    else{
        printf("Client ou médicament inexistant !!!\n");
    }
}

void afficherPharmacie(Pharmacie ph){
    // Question 2.f
    int i;
    for(i=0;i<ph.nbC;i++){
        printf("Client %s a un credit %.2f\n",
            ph.tabC[i].nom, ph.tabC[i].credit);
    }
    printf("-----\n");
    // Question 2.g
    for(i=0;i<ph.nbM;i++){
        printf("Médicament %s %.2f %d\n",
            ph.tabM[i].libelle, ph.tabM[i].prix, ph.tabM[i].qteStock);
    }
    printf("-----\n");
    printf("Solde de la Pharmacie est : %.2f\n", ph.solde);
    printf("-----\n");
}

void enregistrerFichClient(Client c, char fichClient[]){
    FILE * pf;
    pf = fopen(fichClient, "ab+");
    if(pf==NULL){
        printf("Impossible d'ouvrir le fichier %s\n", fichClient);
        exit(1);
    }
    fwrite(&c, sizeof(Client), 1, pf);
    fclose(pf);
}

int nombreClient(char fichClient[]){
    FILE * pf;
    pf = fopen(fichClient, "ab+");
    if(pf==NULL){
        printf("Impossible d'ouvrir le fichier %s\n", fichClient);
        exit(1);
    }
    fseek(pf, 0, SEEK_END);
    int pos = ftell(pf);
    return pos/sizeof(Client);
}

void listerClients(char fichClient[]){
    FILE * pf;
    pf = fopen(fichClient, "ab+");
    if(pf==NULL){
        printf("Impossible d'ouvrir le fichier %s\n", fichClient);
    }
}

```

```
        exit(1);
    }
    Client * tabC;
    int nb = nombreClient(fichClient);
    tabC = (Client*)malloc(nb * sizeof(Client));
    fread(tabC, sizeof(Client), nb, pf);
    int i;
    for(i=0;i<nb;i++){
        printf("Client : %s\t%.2lf\n", tabC[i].nom, tabC[i].credit);
    }
    fclose(pf);
}

void lireClients(char fichClient[], Client ** tabC){
    FILE * pf;
    pf = fopen(fichClient, "ab+");
    if(pf==NULL){
        printf("Impossible d'ouvrir le fichier %s\n", fichClient);
        exit(1);
    }
    int nb = nombreClient(fichClient);
    (*tabC) = (Client*)malloc(nb * sizeof(Client));
    fread((*tabC), sizeof(Client), nb, pf);
    fclose(pf);
}

main(){
    Client c1, c2;

    initClient(&c1, "Hamid", 300);
    initClient(&c2, "Aicha", -400);

    /*enregistrerFichClient(c1, "clients.data");
    enregistrerFichClient(c2, "clients.data");*/
    int nb = nombreClient("clients.data");
    printf("Le nombre de client est : %d\n", nb);
    listerClients("clients.data");
    Client * TC;
    lireClients("clients.data", &TC);
    printf("-----\n");
    int i;
    for(i=0;i<nb;i++){
        printf("Client : %s\t%.2lf\n", TC[i].nom, TC[i].credit);
    }
}
```