

## Programmation II : SMI (S4)

### TP N° 3 : Les Fonctions et la récursivité (Correction)

#### Partie I : Exercices préliminaires

##### Exercice 1 :

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

struct Compte{
    long num;
    char client[20];
    double solde;
};
typedef struct Compte Compte;

long next = 100;

void initCompte(Compte * adrCB, char n[], double s){
    adrCB->num = ++next;
    strcpy(adrCB->client, n);
    adrCB->solde = s;
}

void crediter(Compte * adrCB, double som){
    adrCB->solde += som;
}

int debiter(Compte * adrCB, double som){
    if(som <= adrCB->solde){
        adrCB->solde -= som;
        return 1;
    }
    return 0;
}

int virer(Compte * adrCBSr, Compte * adrCBDest, double som){
    if(debiter(adrCBSr, som) == 1){
        crediter(adrCBDest, som);
        return 1;
    }
    return 0;
}

void afficher(Compte cb){
    printf("%s, Compte numero : %ld, Solde : %.2f dh\n", cb.client, cb.num, cb.solde);
}

int main(){
    Compte A, B, *C;
    initCompte(&A, "Alaoui", 15000);
    initCompte(&B, "Fathi", 20000);
    C = (Compte*)malloc(sizeof(Compte));
    initCompte(C, "Karim", 4000);

    printf("Avant les opérations\n");
    afficher(A);
    afficher(B);
    afficher(*C);
}
```

```
crediter(&A, 2000);
debiter(&B, 300);
virer(&A, C, 200);

printf("Après les opérations\n");
afficher(A);
afficher(B);
afficher(*C);
}
```

## 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 enregistrerMedicament(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");
}

main(){
    Client c1, c2;

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

    afficherClient(c1);
    afficherClient(c2);

    Médicament m1, m2, m3;

    initMédicament(&m1, "Cataflame", 49, 14);
    initMédicament(&m2, "Doliprane", 22, 16);
    initMédicament(&m3, "Baycutene", 25, 6);
```

```
    afficherMedicament(m1);
    afficherMedicament(m2);
    afficherMedicament(m3);

    Pharmacie ph;

    initPharmacie(&ph, 30000, 50, 50);

    // Question 2.d
    enregistrerClient(&ph, c1);
    enregistrerClient(&ph, c2);

    // Question 2.e
    enregistrerMedicament(&ph, m1);
    enregistrerMedicament(&ph, m2);
    enregistrerMedicament(&ph, m3);

    // Question 2.f et 2.g
    afficherPharmacie(ph);

    // Question 2.h
    // Achat de 10 unité du médicament m1
    approvisionner(&ph, m1.libelle, 10);

    // Question 2.i
    achat(&ph, c1.nom, m2.libelle, 1);
    printf("-----\n");

    afficherPharmacie(ph);
}
```