

Correction TP5

Structures de données en langage C

Parcours MIPC

Exercice 1 :

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

```
typedef struct medicament {
    char * libelle;
    int quantite;
    float prix;
    struct medicament *sag ;
    struct medicament *sad ;
} Medicament;
```

```
Medicament *Racine;
```

```
initialisation () {Racine=NULL;}
```

```
int Vendre(char *Med, int NbBoites) {
    Medicament *Courant, *Save, *q, *r, *s;
    int t;
    if (Racine == NULL) { printf("pas de médicaments dans la pharmacie \n"); return -1; }
    Courant=Racine; Save=Racine;
    while (Courant!=NULL)
        if (strcmp(Courant->libelle, Med)> 0){Save=Courant; Courant=Courant->sag;}
        else if (strcmp(Courant->libelle, Med)< 0){Save=Courant; Courant=Courant->sad;}
        else break;
    if (Courant==NULL)
        { printf("ce médicament n existe pas dans la pharmacie \n"); return -1; }
    t= Courant->quantite - NbBoites;
    if (t>0) Courant->quantite=t;
    else {
        if (t<0) printf("tu ne peux vendre que %d Boîtes \n",Courant->quantite);
        Courant->quantite=0;
    }
}
```

```

//suppression
if ((Courant->sag==NULL) && (Courant->sad==NULL)) {
    if (strcmp(Save->libelle, Med)> 0) Save->sag=NULL;
    if (strcmp(Save->libelle, Med)< 0) Save->sad=NULL;
    free(Courant->libelle);
    free(Courant);
}
else if (Courant->sag==NULL) { // pas de fils gauche
    if (strcmp(Save->libelle, Med)> 0) Save->sag=Courant->sad;
    if (strcmp(Save->libelle, Med)< 0) Save->sad=Courant->sad;
    free(Courant->libelle);
    free(Courant);
}
else if (Courant->sad==NULL) { // pas de fils droit
    if (strcmp(Save->libelle, Med)> 0) Save->sag=Courant->sag;
    if (strcmp(Save->libelle, Med)< 0) Save->sad=Courant->sag;
    free(Courant->libelle);
    free(Courant);
}
else //deux fils
    Remplacer_Gauche(Courant);
}
return 0;
}

```

```

Remplacer_Gauche(Nœud *q) {
    Nœud *R, *S;
    R=q->sad; S=NULL;
    while (R->sag !=NULL) {S=R; R=R->sag;}
    free(q->libelle);
    q->libelle=R->libelle;
    q->quantite=R->quantite;
    q->prix=R->prix;
    if (S==NULL) q->sad=R->sad; else S->sag= R->sad;
    free(R);
}

```

```

Medicament *preparer_Medicament(char *Med, int NbBoites, float Prix) {
    Medicament *element;
    if ((element = (Medicament *) malloc(sizeof(Medicament)))==NULL) return NULL;
    if ((element->libelle = (char *) malloc(50*sizeof(char)))==NULL) return NULL;
    strcpy(element->libelle, Med);
    element->quantite=NbBoites;
    element->prix=Prix;
}

```

```

element->sag=NULL;
element->sad=NULL;
return element;
}

```

```

int Acheter(char *Med, int NbBoites, float Prix){
Medicament *Courant, *Save, *element;
int t;
if (Racine ==NULL) { // insertion arbre vide
    element=preparer_Medicament(Med, NbBoites, Prix);
    if (element==NULL) { printf("tu ne peux pas ajouter ce nouveau médicament \n");
                        return -1;}

    Racine=element;
    return 0 ;
}
Courant=Racine; Save=Racine;
while (Courant!=NULL) {
    if (strcmp(Courant->libelle, Med)> 0) {Save=Courant; Courant=Courant->sag;}
    else if (strcmp(Save->libelle, Med)< 0) {Save=Courant; Courant=Courant->sad;}
    else break;
}
if (Courant!=NULL) { //Médicament trouvé
    Courant->quantite+=NbBoites;
    if (Courant->prix!=Prix) Courant->prix=Prix;
    return 0 ;
}
element=preparer_Medicament(Med, NbBoites, Prix);
if (element==NULL) {
    printf("tu ne peux pas ajouter ce nouveau médicament \n");
    return -1;
}
if (strcmp(Save->libelle, Med)> 0) Save->sag=element;
if (strcmp(Save->libelle, Med)< 0) Save->sad=element;
return 0;
}

```

```

void PrixStock(Medicament *Courant,float *t) {
    *t=*t + (Courant->quantite * Courant->prix);
    if (Courant->sag !=NULL) PrixStock(Courant->sag,t);
    if (Courant->sad !=NULL) PrixStock(Courant->sad,t);
}

```

```

//float PrixStock(Medicament *Courant){
//static float t;
//    t = t + (Courant->quantite * Courant->prix);
//    if (Courant->sag !=NULL) PrixStock(Courant->sag);
//    if (Courant->sad !=NULL) PrixStock(Courant->sad);
//return t;
//}

```

```

void Afficher(Medicament *Courant) {
    printf("%s \t\t %d \t%f \n",Courant->libelle, Courant->quantite,Courant->prix);
    if (Courant->sag !=NULL) Afficher(Courant->sag);
    if (Courant->sad !=NULL) Afficher(Courant->sad);
}

```

Exercise 2 :

```

#include<stdio.h>

```

```

typedef struct noeud {
    int val;
    int niveau;
    struct noeud *fg;
    struct noeud *fd; } Noeud;

```

```

void initialiser(Noeud *a) {
    a = NULL;
}

```

```

Niveau_Noeud(Noeud *b, int i) {
    b->niveau=i;
    if (b->fg!=NULL) Niveau_Noeud(b->fg, i+1);
    if (b->fd!=NULL) Niveau_Noeud(b->fd, i+1);
}

```