

Correction TD2 MIPC

Structures de Données en Langage C

Exercice 1 :

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

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

```
int NbMedicaments;
Medicament *Debut;
Medicament *Fin;
```

```
initialisation () { NbMedicaments=0; Debut=NULL; Fin=NULL;}
```

```
int Vendre(char *Med, int NbBoites){
    Medicament *Courant, *Save;
    int t;
    if (Debut ==NULL) {printf("pas de médicaments dans la pharmacie \n");return -1;}
    Courant=Debut; Save=Debut;
    while (Courant!=NULL)
        if (strcmp(Courant->libelle, Med)!=0){Save=Courant; Courant=Courant->suivant;}
        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==Debut) Debut=Debut->suivant;
if (Debut==NULL) Fin = NULL;
else {
    Save->suivant=Courant->suivant;
    if (Save->suivant==NULL) Fin=Save;
    NbMedicaments--;
}
free(Courant->libelle);
free(Courant);
}
return 0;
}

```

```

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->suivant=NULL;
return element;
}

```

```

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

```

```

if (Debut ==NULL) { //insertion liste vide
    Debut=element;
    Fin=element;
    NbMedicaments++;
}
else {
    Courant=Debut;
    while (Courant!=NULL)
        if (strcmp(Courant->libelle, Med)!=0) Courant=Courant->suivant;
        else break;
}

```

```

    if (Courant==NULL) { //insertion fin (médicament inexistant)
        Fin->suivant=element;
        Fin=element;
        NbMedicaments++;
    }
    else { // mise à jour de la quantité (médicament existant)
        Courant->quantite=Courant->quantite+NbBoites;
        if (Courant->prix !=Prix) Courant->prix=Prix;
    }
}
return 0;
}

```

```

float PrixStock(){
Medicament *Courant;
float t=0;
if (Debut ==NULL) { printf("Le stock est vide \n"); return 0;}
else {
    Courant=Debut;
    while (Courant!=NULL){
        t=t + (Courant->quantite * Courant->prix);
        Courant=Courant->suivant;
    }
    return t;
} // else
}

```

```

Afficher() {
Medicament *Courant;
if (Debut ==NULL) printf("Le stock est vide \n");
else {
    Courant=Debut;
    while (Courant!=NULL){
        printf("%s \t\t %d \t%f \n",Courant->libelle, Courant->quantite,Courant->prix);
        Courant=Courant->suivant;
    }
}
}

```