



UNIVERSITÉ MOHAMMED PREMIER OUJDA
FACULTÉ PLURIDISCIPLINAIRE DE NADOR

Science Mathématique Informatique
(SMI)

Rapport

Mini projet Langage C

Réalisé par :

Bilal Ben Aouad

Année universitaire : 2020/2021

Table des matières

1	Explication de Code :	4
1.1	Creation de type des donnees qui va stockées dans la liste :	4
1.2	Creation de liste :	5
1.3	La fonction d'ajoute un element au liste :	5
1.4	La fonction d'affichage d'elements de liste :	6
1.5	La fonction menu :	6
1.6	La fonction de choix :	7
1.7	La fonction de lancemnt d'operation choisi : choix 1	7
1.8	La fonction de lancemnt d'operation choisi : choix 2	8
1.9	La fonction de lancemnt d'operation choisi : choix 3	8
1.10	La fonction de lancemnt d'operation choisi : choix 4	9
1.11	La fonction de lancemnt d'operation choisi : default	9
1.12	fonction de retour	10
1.13	fonction d'affichage les caracteristiques d'un employee trouvé	10
1.14	appelé des fonction dans le main().	11
2	Execution de code :	12
2.1	L'interface graphique :	12
2.2	Operation d'ajoute d'un employee :	13
2.3	Operation d'affichage d'un employee :	13
2.4	Operation de rechercher un employee par le nom et affiche ses caracteris- tiques :	13
2.5	Operation d'initialiser la liste :	14
2.6	liste apres l'initialisation :	14

Liste des figures

1.1	structure de type employee	4
1.2	la structure de creation de liste	5
1.3	fonction d'ajoute a la fin de liste	5
1.4	fonction d'affichage de contenu de liste	6
1.5	la fonction menu	6
1.6	la fonction qui recoit le choix	7
1.7	operation 1 : Ajoute d'un employe	8
1.8	operation 2 : affiche les employees	8
1.9	operation 3 :verification d'existence d'un nom dans la liste	9
1.10	operation 4 :Initialisé la liste	9
1.11	operation 5 :message d'erreur	10
1.12	operation 5 :message de retourner au menu	10
1.13	affichage les caracteristiques d'un employee	10
1.14	declaration des fonction dans la main()	11
2.1	le menu	12
2.2	interface d'ajoute un employee	13
2.3	interface d'affichage d'un employee	13
2.4	interface de rechercher d'un employee et affiche ses informations	13
2.5	interface d'initialisation de liste	14
2.6	le message apparait apres l'initialisation de liste	14

Introduction

Pour stocker des données en mémoire, nous avons utilisé des variables simples (type `int`, `double`...) , des tableaux et des structures personnalisées. Si vous souhaitez stocker une série de données, le plus simple est en général d'utiliser des tableaux.

Toutefois, les tableaux se révèlent parfois assez limités. Par exemple, si vous créez un tableau de 10 cases et que vous vous rendez compte plus tard dans votre programme que vous avez besoin de plus d'espace, il sera impossible d'agrandir ce tableau. De même, il n'est pas possible d'insérer une case au milieu du tableau.

Les listes chaînées représentent une façon d'organiser les données en mémoire de manière beaucoup plus flexible. Comme à la base le langage C ne propose pas ce système de stockage, nous allons devoir le créer nous-mêmes de toutes pièces. Ce exercice qui nous va faire dans ce mini projet nous va aider à être plus à l'aise avec le langage.

Theme de ce TP

Exercer sur l'utilisation des listes chaînées ,les structure et les fonctions.

Objectif de ce TP

- **Créer une liste chaînée .**
- **.Ajouter un employé dans la liste .**
- **Afficher la liste.**
- **Rechercher un employé dans la liste et affiche son caractéristique.**

1

Explication de Code :

1.1 Creation de type des donnees qui va stockées dans la liste :

Dans cet exercice on est demandé d'ajouter des employees a une liste ,chaque employé est défini par un nom, un prénom et un salaire,alors il faut creer un type d'employé qui contient tous les information precedent . on peut faire ça grâce à la structure suivant :



```
typedef struct employe
{
    char nom[10];
    char prenom[10] ;
    float salaire ;
} employe ;
```

FIGURE 1.1 – structure de type employee

1.2 Creation de liste :

le type de cette liste est Box ,il contient un variable de type employe nommé data est un variable pointure de type Box pour gerer le traitement dans la liste .

```
typedef struct Box
{
    employe data ;
    struct Box * next ;
} Box;
```

FIGURE 1.2 – la structure de creaction de liste

1.3 La fonction d'ajoute un element au liste :

Cette fonction est de type Box (c'est le type de liste) ,grace a cette fonction en prend le debut de la liste c'est un pointure et en increment ce pointure jusqu'a on arrive a la din de la liste en suite il cree une cellule a ajoute l'element ,finalment la fonction retourner la nouvelle etat de la liste .

```
Box * ajouterALaFin (Box * debut,employe elm) //debut c'est le debut de la liste ;
{
    Box *b ,*temp ;    // deux pointure pour parcourir le traitement dans le tableau ..
    b=(Box*)malloc(sizeof(Box));
    b->data=elm ;
    b->next =NULL;
    if(debut==NULL)
        debut=b;
    else
    {
        temp=debut ;
        while(temp->next!=NULL)
            temp=temp->next;
        temp->next=b ;
    }
    return debut;
}
```

FIGURE 1.3 – fonction d'ajoute a la fin de liste

1.4 La fonction d'affichage d'elements de liste :

Cette fonction nous a permis d'afficher les element stockees dans la liste d'apres un debut donnee jusqu'a la fin de la liste .

```
void afficherListe(Box * debut )
{
    Box * temp ;
    temp =debut ;
    if(debut==NULL)
    printf("la liste est vide!\n") ;
    else
    {
        while(temp!=NULL) // Tant que le pointeur n'a pas atteint la fin de la liste alors faire
        {
            //afficher les informations des employes enregistré dans la liste
            printf("\nle Nom d'employee Num %d est : %s\n",i,temp->data.nom);
            printf("\nle prenom d'employee Num %d est : %s\n",i,temp->data.prenom);
            printf("\nle salaire d'employee Num %d est : %f\n",i,temp->data.salaire);

            temp=temp->next; // incrémentation de pointure ;

            printf("\n\n\n") ;
        }
    }
}
```

FIGURE 1.4 – fonction d'affichage de contenu de liste

1.5 La fonction menu :

*C'est l'interface graphique qui affiche les operations disponibles .en ajoute la liste dans les parametre de fonction pour les utilise dans les fonction suivantes (car on a une fonction dans une autre...) .dans cette fonction il existe une fonction *valeur()* ; en suite nous verrons le role de cette fonction.*

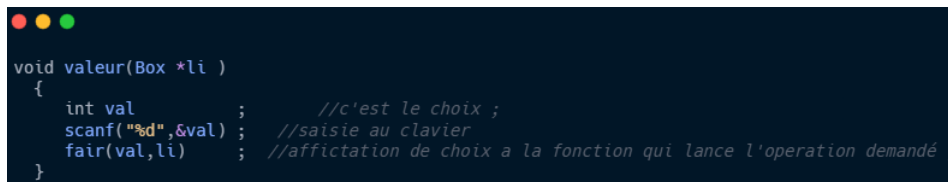
```
void interfaceChoix(Box * li)
{
    printf("\n\n");
    printf("*****\n\n");
    printf("  ++ Gestion De Liste ++\n\n");
    printf("*****\n\n");
    printf("\n");

    printf("  1)- Ajouter un employe   \n\n");
    printf("  2)- Afficher la liste   \n\n"); // interface de menu ;
    printf("  3)- Rechercher un employe \n\n");
    printf("  4)- initialiser la liste \n\n");
    printf("  5)- Quitter             \n\n");
    printf("--> Veuillez choisir une operation a faire :\t") ;
    valeur(li); // affecter la liste sur lequel on travaille a la fonction "valeur()";
}
```

FIGURE 1.5 – la fonction menu

1.6 La fonction de choix :

Cette fonction nous a permis de saisir le choix choisi et l'affecté dans les parametre de la fonction `fair()`; en suite nous verrons le role de cette fonction.

A screenshot of a code editor with a dark blue background and light blue text. The code defines a function named 'valeur' that takes a pointer to a 'Box' structure as an argument. Inside the function, an integer variable 'val' is declared, and 'scanf' is used to read an integer from the keyboard. Finally, the 'fair' function is called with 'val' and the 'li' pointer. Comments in French explain each step: 'c'est le choix', 'saisie au clavier', and 'affectation de choix a la fonction qui lance l'operation demandé'.

```
void valeur(Box *li )
{
    int val ; //c'est le choix ;
    scanf("%d",&val) ; //saisie au clavier
    fair(val,li) ; //affectation de choix a la fonction qui lance l'operation demandé ;
}
```

FIGURE 1.6 – la fonction qui recoit le choix

1.7 La fonction de lancemnt d'operation choisi : choix 1

Cette fonction cointient tous les operations disponibles, il nous a permis de lancer l'operation choisi selon le choix qui nous affecté dans ces parametre dans la fonction `valeur()`; dans ce cas on a l'operation d'ajoute un employee ce for d'un tableau . ainsi dans cette fonction il existe une foctions `reponse()`; en suite nous verrons le role de cette fonction.


```

void fair(int h, Box * li)
{
    switch(h) // h est le choix ;
    {
        case 1 :
        {
            //operation 1 : Ajoute d'un employe ;
            int i ,nbr;

            printf("\n===== Ajoute d'un employe =====\n\n") ;
            printf("Donner le nombre des elements :\t");
            scanf("%d",&nbr);
            employe personne[nbr] ;

            for(i=0;i<nbr;i++)
            {
                printf("\nEntrer le Nom d'employee num %d      :\t",i+1) ;
                scanf("%s",(personne[i].nom)) ;
                printf("\nEntrer le prenom d'employee num %d    :\t",i+1) ;
                scanf("%s",(personne[i].prenom)) ;
                printf("\nEntrer le salaire d'employee num %d    :\t",i+1) ;
                scanf("%f",&(personne[i].salaire)) ;

                li = ajouterALaFin(li,personne[i]) ;
            }

            printf("\nDone!\n");
            if(reponse()==1){interfaceChoix(li);}
            break ;
        }
    }
}

```

FIGURE 1.7 – operation 1 : Ajoute d'un employe

1.8 La fonction de lancemnt d'operation choisi : choix 2

ces le deuxieme cas des operation a faire. c'est le cas d'affichage de contenu de la liste grace a la fonction *afficherListt()* ; .

```

case 2 :
{
    //operation 2 : affiche des employes ;
    printf("\n===== affiche d'un employe =====\n\n") ;
    afficherListe(li); //fonction d'affichage ;
    printf("\nDone!\n");

    if(reponse()==1){interfaceChoix(li);} //pour Quitter ou retour au menu;
    break;
}

```

FIGURE 1.8 – operation 2 : affiche les employes

1.9 La fonction de lancemnt d'operation choisi : choix 3

ces le troisieme cas des operation a faire. c'est le cas de rechercher sur un employee

par son nom la liste . ainsi dans cette fonction il existe une foctions *recherche()* ; en suite nous verrons le role de cette fonction.

```

case 3 :
{
    //operation 3 : rechercher un employees par le nom et affiche les caracteristiques ;
    char cherche[10];
    printf("\nEntrer le nom a rechercher : \t" );
    scanf("%s",cherche);

    if(rechercher(li,cherche)==0 )
    {
        printf("\nAucune information trouve sur le nom que vous entre !\n");
    }
    else
    {printf("\n\n");}

    printf("\nDone!\n");
    if(reponse()==1){interfaceChoix(li);}
    break ;
}

```

FIGURE 1.9 – operation 3 :verification d’existence d’un nom dans la liste

1.10 La fonction de lancemnt d’operation choisi : choix 4

ces le quatrieme cas des operation a faire. c’est le cas d’initialiser la liste .

```

case 4 :
{
    //operation 4 : initialiser la liste e à la liste vide ;
    char rp[7] ;
    if(li==NULL)
    {
        printf("\nLa liste est deja vide !! \n\n");
        if(reponse()==1){interfaceChoix(li);}
    }
    else {
        printf("\nEtes-vous sur de vouloir vider la liste ? (oui/non) :\t ");
        scanf("%s",rp) ;
        if(strcmp(rp, "oui") ==0 || strcmp(rp, "OUI") ==0)
            li =NULL ;
        else printf("\nOK !\n" );
        printf("\nDone!\n");
        if(reponse()==1){interfaceChoix(li);}
        break ;
    }
}

```

FIGURE 1.10 – operation 4 :Initialisé la liste

1.11 La fonction de lancemnt d’operation choisi : default

ces le message qui apparait un message d’erreur lorsque le choix est incorrecte. c’est le cas de quitter le console .

```

default :
{
    // le message sera affiché au cas de le choix choisi n'existe p
    printf("\nChoix incorrecte d(*_*)b\n");
    if(reponse()==1)
    {interfaceChoix(li);}
}
}
}

```

FIGURE 1.11 – operation 5 :message d'erreur .

1.12 fonction de retour

ces le message qui nous a permis de retourner au menu.

```

int reponse()
{
    char r[7];
    printf("\nRetour au menu? (oui/non) : \t" );
    scanf("%s",&r );
    if(strcmp(r, "oui") ==0 || strcmp(r, "OUI") ==0)//c'est une fonction prédéfini de Comparaison de
    deux mots
    return 1;
    else
    return 0;
}

```

FIGURE 1.12 – operation 5 :message de retourner au menu .

1.13 fonction d'affichage les caracteristiques d'un em- ployee trouvé .

*ces la fonction qui affiche les information d'un employee trouvé d'après la recherche
par le nom.*

```

int rechercher(Box *t, char nomModule[10]){
    Box *ptr = t;

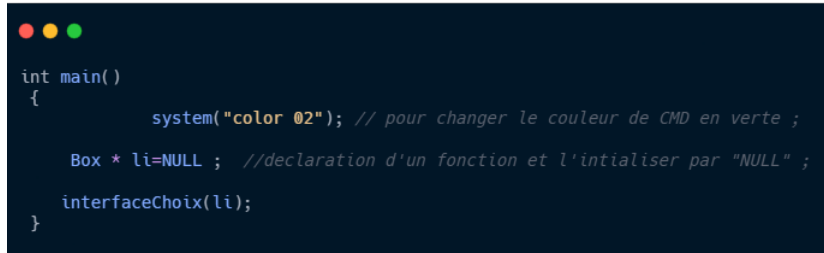
    while(ptr != NULL)
    {
        if(strcmp(ptr->data.nom, nomModule)==0)
        {
            printf("\nLes caracteristiques de * %s * sont : Nom : %s | Prenom : %s | salaire :
%f\n",nomModule,ptr->data.nom,ptr->data.prenom,ptr->data.salaire) ;
            return 1 ;
            ptr =NULL ;
        }
        else
            ptr = ptr->next ;
    }
    return 0 ;
}

```

FIGURE 1.13 – affichage les caracteristiques d'un employee .

1.14 appelé des fonction dans le main().

*on appelle juste une seule fonction dans la main() c'est la fonction `interfaceChoix()` ;
(la fonction d'affichage de menu (toutes les opérations viendront via cette fonction) . et
on declare une liste .*

A screenshot of a code editor with a dark blue background and three colored window control buttons (red, yellow, green) in the top left corner. The code is written in a light blue font and shows the declaration of the main function, including a system call to change the terminal color to green, a pointer declaration for a list, and a call to the interfaceChoix function.

```
int main()
{
    system("color 02"); // pour changer le couleur de CMD en verte ;

    Box * li=NULL ; //declaration d'un fonction et l'initialiser par "NULL" ;

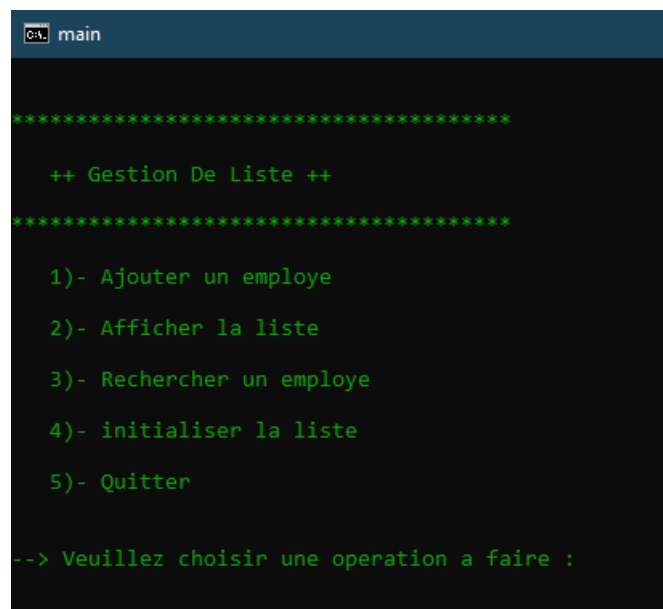
    interfaceChoix(li);
}
```

FIGURE 1.14 – declaration des fonction dans la main() .

2

Execution de code :

2.1 L'interface graphique :



```
main
*****
++ Gestion De Liste ++
*****
1)- Ajouter un employe
2)- Afficher la liste
3)- Rechercher un employe
4)- initialiser la liste
5)- Quitter
--> Veuillez choisir une operation a faire :
```

FIGURE 2.1 – le menu .

2.2 Operation d'ajoute d'un employee :

```
--> Veuillez choisir une operation a faire : 1
===== Ajoute d'un employe =====
Donner le nombre des elements : 1
Entrer le Nom d'employee num 1      : bilal
Entrer le prenom d'employee num 1   : benA
Entrer le salaire d'employee num 1  : 273.50
Done!
Retour au menu? (oui/non) :
```

FIGURE 2.2 – interface d'ajoute un employee .

2.3 Operation d'affichage d'un employee :

```
--> Veuillez choisir une operation a faire : 2
===== affiche d'un employe =====

le Nom d'employee Num 1 est : bilal
le prenom d'employee Num 1 est : benA
le salaire d'employee Num 1 est : 273.500000

Done!
Retour au menu? (oui/non) :
```

FIGURE 2.3 – interface d'affichage d'un employee .

2.4 Operation de rechercher un employee par le nom et affiche ses caracteristiques :

```
--> Veuillez choisir une operation a faire : 3
Entrer le nom a rechercher : bilal
Les caracteristiques de * bilal * sont : Nom : bilal | Prenom : benA | salaire : 273.500000
Done!
Retour au menu? (oui/non) :
```

FIGURE 2.4 – interface de rechercher d'un employee et affiche ses informations .

2.5 Operation d'initialiser la liste :

```
--> Veuillez choisir une operation a faire :    4
Etes-vous sur de vouloir vider la liste ? (oui/non) :    oui
Done!
Retour au menu? (oui/non) :
```

FIGURE 2.5 – interface d'initialisation de liste .

2.6 liste apres l'initialisation :

```
Retour au menu? (oui/non) :    oui

*****
++ Gestion De Liste ++
*****

1)- Ajouter un employe
2)- Afficher la liste
3)- Rechercher un employe
4)- initialiser la liste
5)- Quitter

--> Veuillez choisir une operation a faire :    2
===== affiche d'un employe =====
la liste est vide!
Done!
Retour au menu? (oui/non) :
```

FIGURE 2.6 – le message apparait apres l'initialisation de liste .

Conclusion

Grace à ce Mini Projet ,nous avons pu comprendre et implémenter Les listes chaînées , les structure et les fonction qui nous également aidés à pratiquer l'utilisation de langage C et comprendre son mécanisme de fonctionnement.