



Rapport de mini-projet

“ Gestion des personnels de l’ENSAH ”

Réalisé par :

Mohammed AZIZ

KHADIJA KHOUILI

IKRAM AFIF

Encadré par :

ABDELKHALK EL-BAHRI

Sommaire

-Language c :Historique , avantages,désavantages.....	
-Introduction.....	
A-Gestion de projet.....	
B-Modèle conceptuel	
C-guide de programmeur.....	
1-code complet.....	
2-les structures.....	
3-fonctions et procédures:.....	
3.1-Login(but,idée,code).....	
3.2- Ajoutemp() (but,idée,code).....	
3.3- Afficheemp() (but,idée,code).....	
3.4-modifieremp() (but,idée,code).....	
3.5- Supprimeremp() (but,idée,code).....	
3.6- Consult_profession() (but,idée,code).....	
3.7- chercheremp () (but,idée,code).....	
D-guide d'utilisateur.....	
E-conclusion.....	

Historique

Dans les dernières années, aucun langage de programmation n'a pu se vanter d'une croissance en popularité comparable à celle de C et de son jeune frère C++. L'étonnant dans ce fait est que le langage C n'est pas un nouveau-né dans le monde informatique, mais qu'il trouve ses sources en 1972 dans les 'Bell Laboratories' : Pour développer une version portable du système d'exploitation UNIX, Dennis M. Ritchie a conçu ce langage de programmation structuré, mais très 'près' de la machine.

Avantages

Le grand succès du langage C s'explique par les avantages suivants ; C est un langage :

- C n'est pas orienté vers un domaine d'applications spéciales, comme par exemple FORTRAN(applications scientifiques et techniques) ou COBOL (applications commerciales ou traitant de grandes quantités de données).
- C est basé sur un noyau de fonctions et d'opérateurs limité, qui permet la formulation d'expressions simples, mais efficaces.
- C est un langage structuré, déclaratif et récursif; il offre des structures de contrôle et de déclaration comparables à celles des autres grands langages de ce temps (FORTRAN, ALGOL68,PASCAL).
- Comme C a été développé en premier lieu pour programmer le système d'exploitation UNIX, il offre des opérateurs qui sont très proches de ceux du langage machine et des fonctions qui permettent un accès simple et direct aux fonctions internes de l'ordinateur (p.ex : la gestion de la mémoire).
- comme C permet d'utiliser des expressions et des opérateurs qui sont très proches du langage machine, il est possible de développer des programmes efficaces et rapides.
- bien que C soit un langage près de la machine, il peut être utilisé sur n'importe quel système en possession d'un compilateur C. Au début C' était surtout le langage des systèmes travaillant sous

UNIX, aujourd'hui C'est devenu le langage de programmation standard dans le domaine des micro-ordinateurs.

-en respectant le standard ANSI-C, il est possible d'utiliser le même programme sur tout autre système (autre hardware, autre système d'exploitation), simplement en le recompilant.

-C ne se compose pas seulement des fonctions standard; le langage est animé par des bibliothèques de fonctions privées ou livrées par de nombreuses maisons de développement.

Désavantages

Evidemment, rien n'est parfait. Jetons un petit coup d'œil sur le revers de la médaille :

-En C, nous avons la possibilité d'utiliser des expressions compactes et efficaces. D'autre part, nos programmes doivent rester compréhensibles pour nous-mêmes et pour d'autres. Comme nous allons le constater sur les exemples suivants, ces deux exigences peuvent se contre dire réciproquement.

-La portabilité est l'un des avantages les plus importants de C : en écrivant des programmes qui respectent le standard ANSI-C, nous pouvons les utiliser sur n'importe quelle machine possédant un compilateur ANSI-C. D'autre part, le répertoire des fonctions ANSI-C est assez limité. Si un programmeur désire faire appel à une fonction spécifique de la machine (p.ex : utiliser une carte graphique spéciale), il est assisté par une foule de fonctions 'préfabriquées', mais il doit être conscient qu'il risque de perdre la portabilité. Ainsi, il devient évident que les avantages d'un programme portable doivent être payés par la restriction des moyens de programmation.

- Nous voici arrivés à un point crucial : C est un langage près de la machine, donc dangereux et bien que C soit un langage de programmation structuré, il ne nous force pas à adopter un certain style de programmation (comme p.ex. Pascal).

Dans un certain sens, tout est permis et la tentation de programmer du 'code spaghetti' est grande. (Même la commande 'goto', si redoutée par les puristes ne manque pas en C). Le programmeur a donc beaucoup de libertés, mais aussi des responsabilités : il doit veiller lui-même à adopter un style de programmation propre, solide et compréhensible.

INTRODUCTION

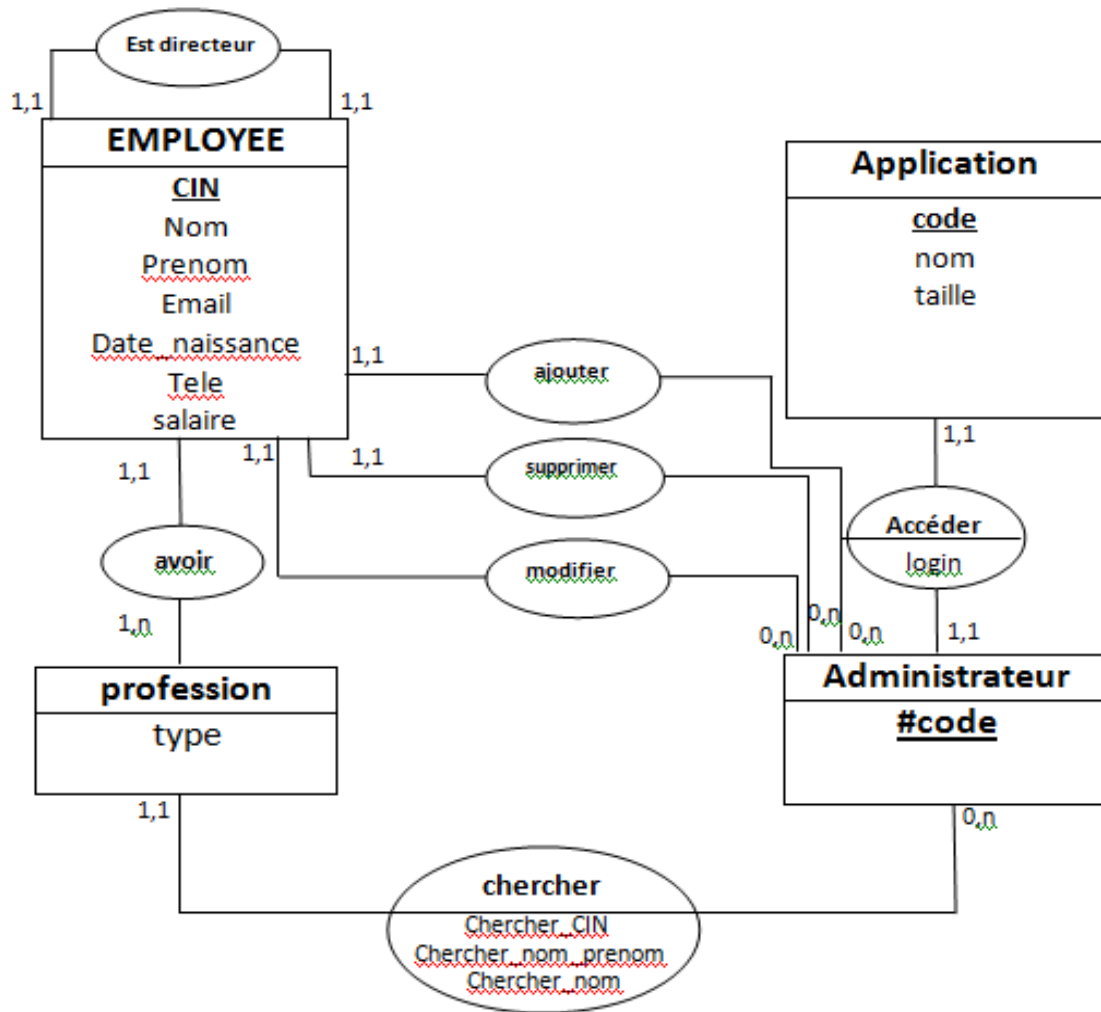
Dans le cadre de notre première année en école d'ingénieurs à L'école nationale des sciences appliquées à Al-Hoceima nous avons eu pour tâche la réalisation d'un projet informatique .Notre objectif était de réaliser une application de gestion des personnels de l'ENSAH en utilisant le langage C. Dans ce dossier nous allons tout d'abord exposer les objectifs de ce projet ,le guide de programmeur puis le guide d'utilisateur .

A-Gestion de projet :

Cette application fournir une Console qui permet à l'utilisateur de l'application de :

- 1-Accéder à l'application..
- 2-Ajouter un personnel .
 - Chaque fonctionnaire se caractérise par : Nom, Prénom, Date de naissance, CIN, email, La profession du personnel, numéro du téléphone, Salaire de base .Vous pouvez ajouter d'autre informations concernant le personnel.
- 3-Supprimer un personnel.
- 4-Modifier un personnel.
- 5-Recherche un personnel par son nom/cin/....
- 6-Afficher la liste des personnels.
- 7-Consulter les membres d'une profession.

B-Modèle conceptuel :



C-Guide de programmeur :

1-les structures :

Pour Cette application ,on aura besoin d'une structure emp a 8 elements .Et ceux pour contenir le nom,le prénom,le CIN,email,la date de naissance,le numéro de téléphone ,le salaire et la profession du personnel,la definition de la structure doit ce faire impérativement avant la fonction main et doit etre suivie d un point virgule après accolade fermante.

Une fois la structure definie,on peut utiliser le type qu'on a construit struct emp pour déclarer les personnels.

2-fonctions :

2.1-Login() :

BUT :

Cette fonction permet d'accéder à l'application à l'aide d'un code.

IDEE :

Cette fonction demande à l'utilisateur d'entrer le code qui permet d'accéder à l'application(3 chances ,l'utilisateur a le droit d'échouer seulement 3 fois) ,si le code est incorrect , l'application donne une autre occasion .sinon il a le droit d'y accéder .

CODE :

```
int login() {
    char code[10];
    int v = 0, i;
    for (i = 0; i < 3; i++) {
        printf("donner le code :\t");
        gets(code);
        if (strcmp(code, "2021A") != 0 && i != 2)
            printf("\nerreur ressayer encore une fois\n");
        if (strcmp(code, "2021A") == 0) {
            v = 1;
            break;
        }
    }
    if (v == 0)
        printf("vous avez echouer dans les trois chances donnee .");
    return v;
}
```

2.2 ajoutemp() :

BUT :

Cette fonction permet d'ajouter un nouveau employée .

IDEE :

La première étape est de placer la curseur à la fin du fichier puis demander à l'utilisateur de saisir les données du personnel (nom,prénom,...),la contrainte qu'on peut rencontrer c'est qu'on sait qu'une établissement a un seul directeur, donc il faut lire la profession au clavier, vérifier si il s'agit de 'directeur', si oui , créer un nouveau variable de type emp, retourner au

début du fichier et commencer à lire les enregistrements, si la profession est semblable à celle dans l'enregistrement, l'application va afficher le message 'directeur existe déjà', puis le traitement va s'arrêter. Sinon, si l'utilisateur saisie une autre profession, elle sera stockée au fichier sans problèmes.

CODE :

```
void ajoutemp() {
    system("cls");
    struct emp e1;
    fseek(fp, 0, SEEK_END);
    another = 'y';
    while (another == 'y' || another == 'Y')
    {
        printf("\nEntrer le nom: ");

        fflush(stdin);
        gets(e.nom);
        printf("\nEntrer le prenom: ");

        gets(e.prenom);
        printf("\nEntrer le CIN: ");

        gets(e.CIN);
        printf("\nEntrer E-MAIL: ");
        gets(e.email);
        printf("\nEntrer la date de naissance: jj/mm/aaaa \n respectez la structure svp noubliez pas de taper /\t");
        scanf("%d/%d/%d", &e.d.jj, &e.d.mm, &e.d.aaaa);
        do {
            printf("\nEntrer le telephone: ");

            fflush(stdin);
            gets(e.tele);
        } while (strlen(e.tele) > 10);
        printf("\nEntrer le salaire: ");
        scanf("%f", &e.salaire);
        do {
            printf("\nEntrer la profession: directeur / professeur / securite/ officier / secretaire:\t");
            scanf("%s", e.profession);
            if (strcmp(e.profession, "directeur") == 0) {
                rewind(fp);
                while (fread(&e1, sizeof(e1), 1, fp) == 1) {
                    if (strcmp(e.profession, e1.profession) == 0) {
                        puts("il existe deja un directeur !!");
                        break;
                    }
                }
            }
        } while ((strcmp(e.profession, "professeur") != 0) && (strcmp(e.profession, "directeur") != 0) && (strcmp(e.profession, "securite") != 0) && (strcmp(e.profession, "officier") != 0) && (strcmp(e.profession, "secretaire") != 0));
        fwrite(&e, sizeof(e), 1, fp);
        printf("\najouter un nouveau employee(y/n) ");
        fflush(stdin);
        another = getche();
    }
}
```

2-3 Afficheemp() :

BUT :

Afficher les données des personnels qui sont déjà enregistrés dans le fichier(dat.bin).

IDEE :

l'application vérifie d'abord si le fichier est non vide,dans ce cas là il affiche les enregistrements existantes.

CODE :

```

void afficheemp(){
    system("cls");
    rewind(fp);
    while(fread(&e,sizeof(e),1,fp)==1) /// read the file and fetch the record one record per fetch
    {
        printf("\nnom:%10s prenom:%10s CIN:%8s EMAIL:%25s DN:%d/%d/%d tele:%10s SALAIRE:%.2f PROFESSION:%10s",
            e.nom,e.prenom,e.CIN,e.email,e.d.jj,e.d.mm,e.d.aaaa,e.tele,e.salaire,e.profession); /// print the name, age and basic salary
    }
    getch();
}

```

2.4- modifieremp() :**BUT :**

Cette fonction permet à l'administrateur de modifier les différents propriétés des employés déjà enregistrés (par choisir un numéro des champs numérotés de 1 à 8 qu'il veut le modifier) .

IDEE :

Premièrement , il faut supprimer tous ce qui est écrit dans la console avant en utilisant system("cls") pour mieux de visibilité ,après le programme demande à l'administrateur d'entrer CIN d'employé à modifier ,et pour cela, le curseur doit déplacer au début de fichier pour chercher les informations de cet employé par la comparaison de CIN entré et les CINs existants dans le fichier .Après la comparaison , si le programme trouve que le CIN entrée existe, alors il donne la possibilité à l'administrateur de choisir un champ parmi les champs numérotés du 1 à 8(pour le cas du dernier champ qui permet de modifier la profession , si l'administrateur modifie la profession d'un employé à 'directeur',le programme cherche s'il existe déjà un directeur avant l'affectation de la nouvelle profession , au cas où il existe déjà un directeur ,l'application affiche un message qu'il existe déjà un directeur et ne modifie rien) .Sinon, l'application affiche le message : " n'existe pas".

Et en fin, l'application donne la possibilité à l'administrateur soit de faire des autres modifications sur les employés soit de sortir de cette zone de modification.

CODE :

```

void modifier() {
    struct emp e1;
    int champ;
    int LA;
    system("cls");
    another = 'y';
    while (another == 'y' || another == 'Y') {
        LA=0;
        printf("\nMODIFIER\n");
        printf("ENTRER LE CIN DE L'EMPLOYEE: ");
        scanf("%s", CINC);
        rewind(fp);
    }
}

```

```

while (fread( & e, sizeof(e), 1, fp) == 1){
    if (strcmp(e.CIN, CINC) == 0){
        LA=1;
        do {
            printf("\n entrer le champ que vous voulez modifier: \n1-nom\n2-prenom\n3-cin\n4-email\n5-date de naissance(dn)\n 6-tel\n7-salaire\n8-proffession \t");
            scanf("%d", & champ);
        } while (champ < 1 && champ > 8);
        switch (champ){
            case 1:{
                printf("\n ancienne nom est %s", e.nom);
                printf("\n entrer le nouveau nom\t");
                fflush(stdin);
                gets(e.nom);
                break;
            }
            case 2:{
                printf("\n ancienne prenom est %s", e.prenom);
                printf("\n entrer le nouveau prenom \t");
                fflush(stdin);
                gets(e.prenom);
                break;
            }
            case 3 :{
                printf("\n ancienne CIN est %s", e.CIN);
                printf("\n entrer le nouveau CIN\t");
                fflush(stdin);
                gets(e.CIN);
                break;
            }
        }
    }
}

```

```

        case 4 :{
            printf("\n ancienne mail est %s", e.email);
            printf("\n entrer le nouveau email\t");
            fflush(stdin);
            gets(e.email);
            break;
        }
        case 5 :{
            printf("\n ancienne date de naissance est %d/%d/%d", e.d.jj, e.d.mm, e.d.aaaa);
            printf("\n entrer le nouveau date de naissance jj/mm/aaaa\t");
            scanf("%d/%d/%d", &e.d.jj, &e.d.mm, &e.d.aaaa);
            break;
        }
        case 6 :{
            printf("\n ancienne telephone est %s", e.tele);
            printf("\n entrer le nouveau telephone");
            fflush(stdin);
            gets(e.tele);
            break;
        }
        case 7 :{
            printf("\n ancienne salaire est %f", e.salaire);
            printf("\n entrer le nouveau salaire");
            fflush(stdin);
            scanf("%f", e.salaire);
            break;
        }
        case 8 :{
            printf("\n ancienne profession est %s", e.proffession);
            printf("\n entrer le nouveau profession :directeur / professeur/ securite/ officier / secretaire \t");
            fflush(stdin);
            gets(e.proffession);
            if (strcmp(e.proffession, "directeur") == 0) {
                rewind(fp);
                while (fread( & e1, sizeof(e1), 1, fp) == 1) {
                    if (strcmp(e.proffession, e1.proffession) == 0) {
                        puts("il existe deja un directeur !!");
                        break;
                    }
                }
            }
            break;
        }
        fseek(fp, -g, SEEK_CUR);
        fwrite( & e, sizeof(e), 1, fp);
        break;
    }
    if(LA==0) printf("\n n'existe pas");
    printf("\n Modifier un autre employee(y/n)");
    fflush(stdin);
    another = getche();
}
}

```

2.5- Supprimeremp() :

BUT :

Supprimer l'employé recherché .

IDEE :

On commence par system("cls")qui sert à effacer tout ce qu'il y a à l'écran, puis on demande à l'utilisateur d'entrer le CIN de personnel recherché. Après on vérifie si ce CIN existe en Comparant successivement les CIN du fichier

avec le CIN donnée. Le nouveau fichier est créé en copiant tous les enregistrements de l'ancien fichier qui précèdent

l'enregistrement à supprimer et tous ceux qui le suivent et en fin on Renommer le fichier temporaire au nom de fichier d'origine.

CODE :

```
void supprimer() {

    system("cls");
    int j=0;
    another = 'y';
    while (another == 'y' || another == 'Y') {
        printf("Enter le CIN de l'employee que vous voulez supprimer: ");
        scanf("%s", CINc);
        ft = fopen("Temp.dat", "wb+");
        rewind(fp);
        j = 0;
        while (fread( & e, sizeof(e), 1, fp) == 1)
        {
            if (strcmp(e.CIN, CINc) != 0)
            {
                fwrite( & e, sizeof(e), 1, ft);
            } else {
                j++;
            }
        }

        fclose(fp);
        fclose(ft);
        remove("EMP.DAT");
        rename("Temp.dat", "EMP.DAT");
        fp = fopen("EMP.DAT", "rb+");
        if (j == 0) {
            puts("Ce CIN n'existe pas .");
        }
        else {
            printf("\n enregistrement supprimée\n");
        }
        printf("Delete another record(y/n)");
        fflush(stdin);
        another = getche();
    }
}
```

2.6- Consult_profession() :

BUT :

Le but de cette fonction est de savoir tous les informations des personnels qui ont la même profession.

IDEE :

L'utilisateur entre la profession au clavier, le curseur se déplace au début, lire les éléments du fichier .si la profession saisis par l'utilisateur est semblable à celle qui existe dans l'élément ,il va l'afficher .Sinon il va lire le deuxième élément, et passer par les mêmes étapes précédentes. Un suit de suite jusqu'à l'arrivée à la fin du fichier.

CODE :

```
void consulterprof() {
    int m=0;
    system("cls");
    do {
        printf("\nEntrer la profession: directeur / professeur / securite/ officier / secretaire:\t");
        fflush(stdin);
        gets(proff);
    } while ((strcmp(proff, "professeur") != 0) && (strcmp(proff, "directeur") != 0) && (strcmp(proff, "securite") != 0) && (strcmp(proff, "officier") != 0) && (strcmp(e.profession, "secretaire") != 0));
    rewind(fp);
    while (fread(& e, sizeof(e), 1, fp) == 1)
    {
        if (strcmp(e.profession, proff) == 0) {
            m++;
            printf("\nnom:%10s   prenom:%10s   CIN:%8s   EMAIL:%25s   DN:%d/%d/%d   tele:%10s   SALAIRE:%.2f   PROFESSION:%10s", e.nom, e.prenom, e.CIN, e.email, e.d.jj, e.d.mm, e.d.aaaa, e.tele, e.salaire, e.prof);
        }
    }
    if(m==0) printf("\nnaucun employee dans la profession %s\n",proff);
    getch();
}
```

2-7-chercheremp() :

Cette fonction est composée de 3 autres fonctions :

2.7-1-cherchernomprenom() :

BUT :

Le but de cette fonction est de chercher les personnels selon le nom et le prénom.

IDEE :

L'utilisateur saisis le nom et prénom du personnel qu'il veut chercher, ils seront stockés sur 2 variables de types char, le curseur se déplace au début du fichier, commencer à lire les éléments un par un, si le nom et prénom

saisis par l'utilisateur est semblable à celles de l'élément, premièrement, il va donner au variable j la valeur 1, puis afficher les données correspondantes. Sinon il va afficher le message 'n'existe pas'.

Remarque : le rôle du variable j est d'éviter la répétition du message 'n'existe pas'.

CODE :

```
void cherchernomprenom() {
    int j = 0;
    system("cls");
    printf("\nEnter le nom de l'employee que vous voulez chercher:\t");
    scanf("%s", empnom);
    printf("\nEnter le prenom de l'employee que vous voulez chercher:\t");
    scanf("%s", empprenom);

    rewind(fp);
    while (fread(&e, sizeof(e), 1, fp) == 1)
    {
        if (strcmp(e.nom, empnom) == 0 && strcmp(e.prenom, empprenom) == 0) {
            j = 1;
            printf("\nnom:%6s    prenom:%6s CIN:%8s EMAIL:%20s DN:%d/%d/%d tele:%10s SALAIRE:%.2f PROFESSION:%10s", e.nom, e.prenom, e.CIN, e.email, e.d.jj, e.d.mm, e.d.aaaa, e.tele, e.salaire);
        }
    }
    if (j == 0) printf("l'employee n'existe pas dans la liste ");
    getch();
}
```

2.7-2-cherchernom() :

BUT :

Le but de cette fonction est de chercher les personnels selon le nom.

IDEE :

L'utilisateur saisis le nom du personnel qu'il veut chercher, ils sera stockés sur 1 variable de type char, le curseur se déplace au début du fichier, commencer à lire les éléments un par un, si le nom saisis par l'utilisateur est semblable à celle de l'élément, premièrement, il va donner au variable a la valeur 1, puis afficher les données correspondantes. Sinon il va afficher le message 'n'existe pas'.

Remarque : le rôle du variable (a) est d'éviter la répétition du message 'n'existe pas'.

CODE :

```

void cherchernom() {
    int a = 0;

    system("cls");
    printf("\nEnter le nom de l'employee que vous voulez chercher:\t");
    scanf("%s", empnom);
    rewind(fp);
    while (fread( & e, sizeof(e), 1, fp) == 1)
    {
        if (strcmp(e.nom, empnom) == 0) {
            a = 1;
            printf("\nnom:%6s prenom:%6s CIN:%8s EMAIL:%20s DN:%d/%d/%d tele:%10s SALAIRE:%.2f PROFESSION:%10s", e.nom, e.prenom, e.CIN, e.email, e.d.jj, e.d.mm, e.d.aaaa, e.tele, e.salaire,
            );
        }
    }
    if (a == 0) printf("l'employee n'existe pas dans la liste ");
    getch();
}

```

2.7-3-cherchecin() :**BUT :**

Le but de cette fonction est de chercher les personnels selon le CIN.

IDEE :

L'utilisateur saisis le CIN du personnel qu'il veut chercher, ils sera stockés sur 1 variable de type char, le curseur se déplace au début du fichier, commencer à lire les éléments un par un, si le CIN saisis par l'utilisateur est semblable à celui de l'élément, premièrement, il va donner au variable b la valeur 1, puis afficher les données correspondantes. Sinon il va afficher le message 'n'existe pas'.

Remarque : le rôle du variable (b) est d'éviter la répétition du message 'n'existe pas'.

CODE :

```

void cherchecin() {
    int b = 0;
    system("cls");
    printf("\nEnter le CIN de l'employee que vous voulez chercher:\t");
    scanf("%s", CINc);
    rewind(fp);
    while (fread( & e, sizeof(e), 1, fp) == 1)
    {
        if (strcmp(e.CIN, CINc) == 0) {
            b = 1;
            printf("\nnom:%6s prenom:%6s CIN:%8s EMAIL:%20s DN:%d/%d/%d tele:%10s SALAIRE:%.2f PROFESSION:%10s", e.nom, e.prenom, e.CIN, e.email, e.d.jj, e.d.mm, e.d.aaaa, e.tele, e.salaire,
            );
        }
    }
    if (b == 0) printf("l'employee n'existe pas dans la liste ");
    getch();
}

```

D-Guide d'utilisateur :

1- Interface d'entrée :

```

#####
#####
#####      application qui gere les personnels de ENSAH      #####
#####
#####
#####
-----
-----

donner le code :
```

-L'utilisateur doit saisir correctement le code de l'application '2021A' pour qu'il peut y accéder.

2- Interface des choix :

```

#####
#####
#####      (:BIENVENUE:)      #####
#####
#####
#####
-----
-----

1. ajouter un employee
2. la liste des employees
3. Modifier un employee par CIN
4. supprimer un employee
5. consulter les membres d'une profession
6. chercher
7. quitter
votre choix:
```


-L'utilisateur doit choisir l'option qu'il veut en tapant un entier de 1->7. (Chaque entier représente une et une seule option.

2-1- ajouter un employé :

```

Entrer le nom: EL BAHRI
Entrer le prenom: ABDELKHALEK
Entrer le CIN: GG111111
Entrer E-MAIL: ELBAHRI@GMAIL.COM
entrer la date de naissance: jj/mm/aaaa
respectez la structure svp noubliez pas de taper / 10/10/2000
Entrer le telephone: 0717171717
Enter le salaire: 10000
Entrer la proffession: directeur / professeur / securite/ officier / secretaire: professeur
ajouter un nouveau employee(y/n)

```

L'utilisateur entre les informations de l'employé (nom ,prénom , ...) , ces derniers seront stockées dans le fichier.

2-2-la liste d'employés :

nom:	AZIZ	prenom:	KAMAL	CIN:	FA192714	EMAIL:	AZIZmbu@gmail.com	DN:	13/11/2000	tele:	0616815066	SALAIRE:	9000.00	PROFESSION:	officier
nom:	AFIF	prenom:	IKRAM	CIN:	KAMA1	EMAIL:	AFIFIKRAM@gmail.com	DN:	10/10/2000	tele:	0818181818	SALAIRE:	9000.00	PROFESSION:	securite
nom:	AZIZ	prenom:	MOHAMMED	CIN:	FA192714	EMAIL:	AZIZmbu@gmail.com	DN:	13/11/2000	tele:	0616815066	SALAIRE:	9000.00	PROFESSION:	directeur
nom:	KHADIJA	prenom:	MOHAMMED	CIN:	GA122122	EMAIL:	AZIZMOHAM@gmail.com	DN:	14/11/1999	tele:	0616161616	SALAIRE:	9000.00	PROFESSION:	professeur
nom:	AZIZ	prenom:	MOHAMMED	CIN:	FA192714	EMAIL:	AZIZMBU@GMAIL.COM	DN:	13/11/2002	tele:	0716815066	SALAIRE:	9999.00	PROFESSION:	officier
nom:	EL BAHRI	prenom:	ABDKHALK	CIN:	GG111111	EMAIL:	ELBAHRI@GMAIL.COM	DN:	10/10/2000	tele:	0717171717	SALAIRE:	10000.00	PROFESSION:	professeur

-cette option permet d'afficher la liste d'employés

2-3- Modifier un employé par CIN :

-si l'utilisateur veut changer quelques informations d'un employé il suffit qu'il a son CIN . Avec le CIN il peut changer n'importe quelle information.

```

MODIFIER
ENTRER LE CIN DE L'EMPLOYEE': GG111111

  entrer le champ que vous voulez modifier:
1-nom
2-prenom
3-cin
4-email
5-date de naissance(dn)
6-telphone
7-salaire
8-proffession  1

ancienne nom est EL BAHRI
entrer le nouveau nom  KAMAL

Modifier un autre employee(y/n)
  
```

Et voici le résultat obtenu :

nom:	AZIZI	prenom:	KAMAL	CIN:	HA188188	EMAIL:	AZIZmbu@gmail.com	DN:	13/11/2000	tele:	0616815066	SALAIRE:	9000.00	PROFESSION:	officier
nom:	AFIF	prenom:	IKRAM	CIN:	KAMA1	EMAIL:	AFIFIKRAM@GMAIL.com	DN:	10/10/2000	tele:	0818181818	SALAIRE:	9000.00	PROFESSION:	securite
nom:	AZIZ	prenom:	MOHAMMED	CIN:	FA192714	EMAIL:	AZIZmbu@gmail.com	DN:	13/11/2000	tele:	0616815066	SALAIRE:	9000.00	PROFESSION:	directeur
nom:	KHADIJA	prenom:	MOHAMMED	CIN:	GA122122	EMAIL:	AZIZMOHAM@GMAIL.COM	DN:	14/11/1999	tele:	0616161616	SALAIRE:	9000.00	PROFESSION:	professeur
nom:	AZIZ	prenom:	MOHAMMED	CIN:	FA192714	EMAIL:	AZIZMBU@GMAIL.COM	DN:	13/11/2002	tele:	0716815066	SALAIRE:	9999.00	PROFESSION:	officier
nom:	KAMAL	prenom:	ABDKHALK	CIN:	GG111111	EMAIL:	ELBAHRI@GMAIL.COM	DN:	10/10/2000	tele:	0717171717	SALAIRE:	10000.00	PROFESSION:	professeur

2-4- supprimer un employé :

```

Entrer le CIN de l'employee que vous voulez supprimer: KAMA1

enregistrement supprimé
supprimer un autre employee(y/n)
  
```

-si l'utilisateur veut supprimer un employé il suffit qu'il a son CIN . Avec le CIN il peut supprimer n'importe quel employé.

Et voici le résultat obtenu :

```
nom: AZIZI prenom: KAMAL CIN:HA188188 EMAIL: AZIZmbu@gmail.com DN:13/11/2000 tele:0616815066 SALAIRE:9000.00 PROFESSION: officier
nom: AZIZ prenom: MOHAMMED CIN:FA192714 EMAIL: AZIZmbu@gmail.com DN:13/11/2000 tele:0616815066 SALAIRE:9000.00 PROFESSION: directeur
nom: KHADIJA prenom: MOHAMMED CIN:GA122122 EMAIL: AZIZMOHAM@gmail.com DN:14/11/1999 tele:0616161616 SALAIRE:9000.00 PROFESSION:professeur
nom: AZIZ prenom: MOHAMMED CIN:FA192714 EMAIL: AZIZMBU@gmail.com DN:13/11/2002 tele:0716815066 SALAIRE:9999.00 PROFESSION: officier
nom: KAMAL prenom: ABDKHALK CIN:GG111111 EMAIL: ELBAHRI@gmail.com DN:10/10/2000 tele:0717171717 SALAIRE:10000.00 PROFESSION:professeur
```

L'employé qui a comme CIN= 'KAMA1' a été bien supprimé de la liste.

2-5- Consulter les employés d'une profession :

```
Entrer la profession: directeur / professeur / securite/ officier / secretaire: professeur
nom: KHADIJA prenom: MOHAMMED CIN:GA122122 EMAIL: AZIZMOHAM@gmail.com DN:14/11/1999 tele:0616161616 SALAIRE:9000.00 PROFESSION:professeur
nom: KAMAL prenom: ABDKHALK CIN:GG111111 EMAIL: ELBAHRI@gmail.com DN:10/10/2000 tele:0717171717 SALAIRE:10000.00 PROFESSION:professeur
```

Si l'utilisateur veut savoir les informations des employés qui travaillent dans une profession, il suffit qu'il entre la profession une parmi ('directeur','professeur','sécurité','officier','secrétaire').

2-6-chercher un employé :

```
donner votre choix :
1-chercher par nom
2-chercher par nom et prenom
3-chercher par CIN
```

Supposons qu'on a choisi de chercher par nom !

Voici le résultat obtenu :

-si l'utilisateur veut chercher un employé, il a 3 options (chercher par

```
Enter le nom áde l'employee que vous voulez chercher': AZIZ
nom: AZIZ prenom:MOHAMMED CIN:FA192714 EMAIL: AZIZmbu@gmail.com DN:13/11/2000 tele:0616815066 SALAIRE:9000.00 PROFESSION: directeur
nom: AZIZ prenom:MOHAMMED CIN:FA192714 EMAIL: AZIZMBU@GMAIL.COM DN:13/11/2002 tele:0716815066 SALAIRE:9999.00 PROFESSION: officier
```

CIN, par nom et prénom, par nom) , donc il suffit de choisir par quel option il veut chercher .

2-7-quitter :

-Cette option permet de quitter l'application.

E-conclusion :

A l'aide de ce projet nous avons pu comprendre et expérimenter les différentes étapes de la conception d'une application. De plus la programmation nous a permis d'améliorer nos connaissances du langage C . En plus d'être un projet pédagogique il est aussi ludique et nous a donné beaucoup de liberté dans le code et dans la conception .