



Département Informatique  
Filière SMI4

Rapport de Mini-Projet en Langage C

# « Gestion de location de voiture »

Module :

## Programmation 2

Année universitaire :2019/2020

Sous supervision de :

**Prof BELAQZIZ Salwa**

Fait par : Khaoula Echine

CNE: 17-73840001

# I-Introduction

Comme c'est marqué au page d'accueil c'est que mon Mini-Projet est sous titre « Gestion de Location de voiture », il consiste à développer une application pour la gestion d'une agence de location de voitures.

Mon Mini-Projet est principalement découpé en trois parties :

- \*L'une est la partie '**Gestion Clients**' qui contient les opérations nécessaires pour la gestion des locataires.

- \*La deuxième partie est '**Gestion Voiture**' qui s'occupe de la gestion des voitures d'agence.

- \*Et la troisième partie qui est la plus importante est '**Location**' qui consiste à gérer les locations des voitures par les clients.

Mon but ici est de vous présenter mon Mini-Projet dans son fonctionnement et les différentes étapes et modifications qu'il a subit.

Pour cela je vous décrivons d'abord les menus de chacun de ces parties et le Menu Principal. Puis on passe aux fonctions existantes au fond de chacune des parties.

Je vous souhaite une bonne lecture ,et j'espère que vous appréciez mon travail.

# II-Les Menus

## 1)Menu Principal

La fonction du titre **Menu\_Principal()** a pour but de rédiger le Menu qui s'affiche au premier à l'utilisateur une fois qu'il Lance l'application.

Alors, on a rien assez compliqué, le travail est fait juste par la fonction `printf()`, et on a introduit `(\xda\x4...\xbf\xb3\xd0\x4...\xc0\xb3)` pour encadrer le titre de la fonction par une bordure de trait simple, ainsi qu'on encadre le corps de la fonction par une bordure de double trait en utilisant la fonction **`printf()`** avec `(\xc9\xcd...\xbb\xba...\xbc\xcd...\xc8\xcd...\xc8\xba..)` et les espaces pour mettre le tout au centre du terminal.

Le corps du Menu contient les titres des trois Menus en ajoutant l'option 'Quitter'. Et au dessous on demande d'utilisateur d'entrer son choix qui sera vérifié par la fonction **`switch()`** alors s'il s'agit du premier choix (**choix=1**), l'utilisateur va passer au Menu de 'Location', et s'il s'agit de deuxième choix (**choix=2**), l'utilisateur va passer au Menu de 'Gestion Voitures', et s'il s'agit de troisième choix (**choix=3**), l'utilisateur va passer au Menu de 'Gestion Clients', enfin s'il a choisi le dernier choix (**choix=4**), l'utilisateur va quitter l'application par la fonction `exit(0)`.

Or si l'utilisateur choisit un chiffre qui n'est pas compris entre 1 et 4, un message sera affiché pour lui demander de choisir une autre option dont le chiffre appartient aux chiffres déjà cités.

Cette fonction est la seule qu'on appelle dans le main.

## 2 )Gestion des Voitures

Le Menu du '**Gestion des Voitures**' est affiché par la fonction de titre « **Gestion\_Voitures()** » en choisissant le deuxième choix au Menu Principal (Gestion voitures).

Le travail qu'on a suivi pour réaliser ce Menu est similaire au travail précédent du Menu Principal, ils se diffèrent seulement en titre du Menu qui est « Gestion des voitures », et aux titres des options qui existent au corps du Menu.

Le corps du Menu permet au utilisateur de faire le choix par la fonction **`switch()`** entre cinq options différentes. Alors, s'il choisit le premier choix (**choix=1**), cela lui fait permet d'afficher la liste des voitures qui existent dans l'agence.

Ainsi, si l'utilisateur choisit le deuxième choix (**choix=2**), il a le droit d'ajouter une nouvelle voiture à l'agence, et s'il choisit le troisième choix (**choix=3**) il peut modifier les informations d'une des voitures déjà enregistrées dans l'agence. Or, s'il choisit le quatrième choix (**choix=4**), cela lui donne le droit de supprimer une voiture de l'agence.

Enfin, s'il choisit le dernier choix (**choix=5**), cela lui permet de revenir au Menu Principal. Et comme on a fait auparavant au Menu Principal, si l'utilisateur fait entrer un chiffre

différents des cinq chiffres déjà cités, un message sera affiché a lui pour saisir une option de chiffre compris entre 1 et 5.

### 3) Gestion des Clients

Une fois que l'application est lancé par l'utilisateur, ce dernier a le droit de choisir le troisième choix au Menu Principal (Gestion Clients) qui lui permet de passer au menu du '**Gestion des Clients**' qui sera affiché grâce au fonction « **Gestion\_Clients()** ».

Au sein de cette fonction, on a copié le même code du fonction précédente sauf au lieu de laisser le choix au utilisateur de gérer les voitures, maintenant il a le choix de gérer le client par l'affichage de la liste des clients en choisissant le premier choix (**choix=1**) du corps du Menu.

Ou d'ajouter un client en choisissant le deuxième choix (**choix=2**) , ou de modifier les informations d'un client d'agence par le troisième choix (**choix=3**). Ou de supprimer un client en faisant entrer le chiffre 4 (**choix=4**), ou enfin de revenir au Menu Principal en choisissant le dernier choix (**choix=5**).

Aussi, un message sera affiché au utilisateur s'il a mal choisit le choix convenable. Tout cela est fait évidemment par la fonction **switch()** .

En haut on a le titre du menu qui est « Gestion des Clients ».

### 4) Location d'une voiture

Le Menu '**Location d'une voiture**' est le plus important dans ce Mini-Projet, même si je l'ai laissé le dernier soit en le décrivant ici ou en le codant au menu.c , c'est pour la simple raison c'est qu'il demande plus de travail au niveau de ses sous fonctions que les deux menus précédents.

L'utilisateur de cette application peut accéder à ce Menu en choisissant le premier choix du Menu Principal (Location), Puis il sera devant les cas de choisir cinq options : visualiser une contrat en tapant le premier choix (**choix=1**), ou louer un voiture en tapant le deuxième choix (**choix=2**), ou retourner une voiture qui a déjà louée en faisant entrer le chiffre 3 (**choix=3**), ou modifier les informations d'une contrat en choisissant le quatrième choix (**choix=4**), ou s'il veut revenir au Menu Principal en tapant le dernier choix (**choix=5**), ou s'il a choisi un chiffre autre que ces derniers un message sera affiché qui lui demande de faire entre un chiffre compris entre 1 et 5.

Tout cela est fait au sein du fonction du titre « **Location\_Voiture()** » en suivant les mêmes étapes des menus précédents : en utilisant les deux fonctions **printf()** et **switch()**.

# II-Fonctions Facultatifs

## 1)Chapeau

Dans mon Mini-Projet j'ai créé une fonction de titre « **Chapeau()** » qui a pour rôle de supprimer tout ce qui était déjà affiché en exécution précédente en utilisant la fonction **system("cls")**, et de changer la couleur de la fenêtre d'exécution par la fonction **system("color 70")** : 7 s'agit de la couleur d'arrière plan qui est « Bright white » et 0 c'est la couleur du texte qui est « Black ». En ajoutant le rôle de dessiner en haut de chaque fonction un entête qui est sous la forme d'un tableau d'une seule ligne et de 3 colonnes, encadré par une bordure de double trait en utilisant la fonction **printf()** comme on a fait dans les menus précédents.

Dans la première colonne on a l'heure actuelle sous format (hh:mm:ss) que j'ai fait en utilisant la fonction **\_strtime(bufftime)** avec **bufftime** est une chaîne de caractère que je stocke au sein de l'instruction **printf(..%s...,bufftime);** pour afficher le temps actuel.

Dans la deuxième colonne qui prend beaucoup d'espace on a le titre du Mini-Projet « Location de voitures ».

Dans la troisième colonne on a la date d'aujourd'hui sous la forme (jour:mois:année) qu'on a affiché en utilisant la fonction **printf(....%s...,buffdate);** et la fonction **\_strdate(buffdate)**.

Dans mon code source j'ai fait l'appel de la fonction **Chapeau()** au début de chaque fonction du code.

## 2)Fin

Dans le code source **Menu.c** j'ai créé trois fonctions au début : **Fin1(), Fin2(), Fin3()**.

**Fin1()** : cette fonction est relié au sous fonctions du menu 'Gestion des Voitures'. A la fin de chaque fonction on la fait appel en demandant au utilisateur de saisir 1 pour revenir au Menu Gestion Voiture ou saisir 0 s'il veut quitter l'application.

**Fin2()** :il s'agit du même code du fonction **Fin1()** sauf celle là est relié au Menu 'Gestion des Clients' il laisse l'utilisateur devant deux choix : soit de revenir au menu Gestion des Clients en tapant le chiffre 1, soit de quitter l'application en tapant le chiffre 0.

**Fin3()** :même principe des deux fonctions précédentes, son rôle est de permettre au utilisateur de choisir entre sortir de l'application en tapant le chiffre 0 ou de revenir au menu 'Location des voitures'.

# III-Gestion des Voitures

Comme on avait déjà mentionné que la fonction **Gestion\_Voitures()** affiche le menu Gestion des Voitures dans lequel l'utilisateur a la liberté de choisir l'option qu'il veut et qui sera lancé grâce au sous fonctions du Menu qu'on va les traiter maintenant une par une.

## 1)Saisie voiture()

Dans mon code source j'ai créé cette fonction qui a pour rôle de remplir le fichier «Voiture» par les informations de chaque voiture.

Pour réaliser cette fonction premièrement on a déclaré un pointeur f sur FILE et on a ouvert le fichier en mode d'ouverture 'w' qui permet d'écrire dans le fichier même si ce dernier n'existe pas , il sera créé : **FILE \*f=fopen("voiture.txt","w") ;**

Deuxièmement, on débute par demander au utilisateur de saisir le nombre de voitures puis on lui demande de faire entrer les infos de chacune des voiture :IdVoiture, marque, nom Voiture, couleur, nombre de places, prix de jour de location, si la voiture est louée ou non ;Tout cela sera saisi et lu au sein d'une **boucle For** et suivant le schéma de la **structure Voiture**. Enfin on ferme le fichier par la fonction **fclose(f)**.

Dans mon fichier Voiture, j'ai saisi huit voitures de marque différentes .  
C'est possible de faire enregistrer les informations dans le fichier directement sans avoir besoin créer cette fonction.

## 2)liste voiture()

La fonction **Liste\_Voiture()** nous permet d'afficher la liste des voitures qui sont enregistrées dans le fichier Voiture qu'on a déjà crée par la fonction précédente.

Afin d'afficher la liste des voitures on a besoin d'abord d'ouvrir le fichier en mode d'ouverture de lecture 'r' : **FILE \*liste=fopen('voiture.txt','r')** , puis on a commencé à lire toutes les variables existantes au sein du fichier au tant qu'on a pas arrivé à sa fin , cela est fait en utilisant la boucle while les la fonction fscanf : **while(fscanf(liste," ....",&....) !=EOF) ;** Et puis on commence à afficher les informations de chaque voiture par la fonction printf().

### 3)ajout\_voiture()

Pour ajouter une nouvelle voiture au fichier «voiture» l'utilisateur choisit le deuxième choix du menu Gestion des Voitures puis il passera à une nouvelle fenêtre qui est affiché grâce à la fonction **ajout\_voiture()**.

Le code de cette fonction débute par la demande de l'utilisateur de saisir l'identifiant de la voiture qu'il souhaite ajouter. Puis on ouvre le fichier «voiture» en mode de lecture et écriture '**r+**' et on commence la lecture des informations existantes dans le fichier par la fonction **fscanf()** et la boucle **while** afin de vérifier si la voiture existe déjà dans le fichier ou non.

Tout cela se fait en comparant l'identifiant entré par l'utilisateur avec les identifiants qui sont dans le fichier si on a une égalité on incrémente une variable nommé '**test**' qui était initialisé par 0.

Après on fait un test sur la variable si elle n'a pas changé (càd  $\text{test}=0$ ) cela signifie que la voiture n'existe pas avant dans le fichier et alors on demande d'utilisateur de saisir les informations de la voiture qui seront ajoutées au fichier par la fonction **fprintf()** puis un message sera affiché qui signifie que l'ajout est fait en succès.

Or si la variable test change de valeur (càd  $\text{test}\neq 0$ ) cela signifie que la voiture qu'on souhaite ajouter existe déjà dans le fichier et par suite un message sera affiché pour l'approuver. Enfin on ferme le fichier par la fonction **fclose()** et on fait l'appel à la fonction Fin1() pour revenir au menu Gestion des voitures ou pour quitter l'application selon le choix de l'utilisateur.

### 4)modification\_voiture()

Afin de modifier une ou plusieurs informations d'une voiture déjà enregistrée dans le fichier «voiture» , nous avons écrit cette fonction **modification\_voiture()** en suivant plusieurs étapes :

Premièrement on ouvre le fichier '**voiture**' en mode d'ouverture : lecture '**r**' et on ouvre un nouveau fichier nommé '**nouveau\_voiture**' qui sera créé par le mode '**w**'.

Puis on demande de l'utilisateur de saisir l'identifiant de la voiture qu'il désire modifier, et on fait un test semblable à celui de la fonction qui précède : on commence la lecture du fichier 'voiture' par la boucle **while** et la fonction **fscanf** et à chaque fois qu'on trouve un id égal au id entré par l'utilisateur on change la valeur de la variable '**test**' de 0 à 1, ce qui signifie que la voiture qu'on souhaite modifier existe bien dans le fichier.

Alors si **test=1** on demande de l'utilisateur de faire choisir l'information qu'il veut modifier :1-marque,2-nom,3-couleur,4-nbplaces,5-prixJour,6-EnLocation, or s'il choisit un chiffre supérieur à 6 un msg sera affiché, après l'avoir choisir le chiffre convenable on lui demande de faire entrer la nouvelle information et on stocke le tout dans le nouveau fichier

« nouveau\_voiture». Quand on termine la partie des choix on continue la copie des autres infos existantes dans l'ancien fichier voiture dans le nouveau fichier .

A la fin on ferme les deux fichiers par la fonction **fclose()**, et on supprime l'ancien fichier voiture par la fonction **remove("voiture.txt")** et on renomme le nouveau fichier par le nom d'ancien fichier **rename("nouveau\_voiture.txt","voiture.txt")** et comme ça on obtient un fichier 'voiture' avec des infos bien modifiées. Sans oublier de tester la variable '**test**' s'elle garde la valeur 0 alors un message sera affiché « voiture introuvable ».

## 5)Suppression voiture()

Pour supprimer une voiture de fichier « voiture » en se sert de la fonction

**suppression\_voiture()** dont on a suivi le même principe de la fonction précédente.

D'abord on déclare deux pointeurs FILE, l'un sert à ouvrir le fichier voiture en mode d'ouverture :lecture '**r**', l'autre sert à créer et ouvrir un nouveau fichier «nouveau\_voiture» en mode d'ouverture :écriture '**w**'.

Puis on demande d'utilisateur de faire entrer l'identifiant de la voiture qu'il souhaite supprimer, et par suite on compare l'id entré par les id des voitures enregistrées dans le fichier en utilisant la boucle **while** et la fonction **fscanf()**. En lisant les infos du fichier , chaque voiture a un id différent de celui entré on copie ses infos dans le nouveau fichier, or si on arrive au même id on change seulement la valeur d'une variable '**test**' de 0 à 1 et on affiche un message qui confirme la suppression de la voiture sans copier les infos de la voiture dans le nouveau fichier.

A la fin on ferme les deux fichiers par la fonction **fclose()** et on supprime le fichier « voiture » et on le remplace par le nouveau fichier on changeant le nom de ce dernier : **rename("nouveau\_voiture.txt","voiture.txt")** ; Et on termine par tester la valeur de la variable '**test**' s'elle est égale encore à 0 on affiche un msg au utilisateur qui lui montre que la voiture qu'il souhaite supprimer n'existe même pas dans le fichier voiture.



# IV-Gestion des Clients

Le menu 'Gestion des Clients' est un menu qui contient des options qui permettent à l'utilisateur de gérer les clients enregistrés dans le fichier 'client'. Tout ça se fait suivant des sous fonctions qu'on va traiter maintenant :

## 1)Saisie\_client()

Cette fonction nous permet de remplir le fichier «client» par les informations des clients en demandant d'abord d'utilisateur de faire entrer le nombre des clients. Puis en ouvrant le fichier «client» en mode d'ouverture de création de fichier et d'écriture en même temps

**'w' : FILE \*f=fopen("client.txt","w") ;**

Après on demande d'utilisateur de saisir les informations de chaque client :identifiant /nom/prénom/cin/ adresse/telephone en servant de la boucle For et les fonction **printf(),scanf(), fprintf()** pour stocker les infos dans le fichier puis on ferme enfin le fichier par la fonction **fclose()** .

## 2)liste\_client()

Afin d'afficher la liste des clients enregistrés dans le fichier «client» on a créé la fonction **liste\_voiture()** qui déclare au début un pointeur FILE qui sert à ouvrir le fichier «client» en mode lecture **'r' : FILE \*liste=fopen("client.txt","r")**. Puis on commence la lecture des informations stockées dans le fichier par la fonction **fscanf()** et la boucle **while()** :

**while(fscanf(liste,"....",&...) !=EOF)** ; puis on affiche les informations sur la fenêtre d'exécution par la fonction **printf()**.

Enfin on ferme le fichier par la fonction **fclose()** et on laisse le choix au utilisateur de revenir au menu en tapant 1 ou de quitter l'application en tapant 0 Tout cela se fait en appelant la fonction **Fin2()**.

## 3) ajout\_client()

Pour ajouter un nouveau client au fichier «client» on a écrit cette fonction **ajout\_client()** qui demande au début d'utilisateur de saisir l'identifiant du client qu'il veut ajouter, puis elle ouvre le fichier «client» en mode lecture et écriture **:FILE \*p=fopen("client.txt","r+") ;**

Puis elle commence la lecture du fichier en se servant de la fonction **fscanf()** et la boucle **while** ,à chaque fois qu'on trouve un identifiant égal à celui entré par l'utilisateur, on incrémente une variable **'test'** qui était initialisée par 0 au début.

Après la lecture, on fait le test sur la variable si cette dernière égal à 0 cela signifie que le client qu'on souhaite ajouter n'existe pas déjà dans le fichier et puis on demande de l'utilisateur de saisir les informations de nouveau client suivant la structure 'client'. Ensuite on ajoute ces informations dans le fichier «client» par la fonction **fprintf()** en affichant un message qui approuve que l'ajout est fait en succès,

Or si la variable '**test**' change de valeur cela signifie que le client existe déjà dans le fichier alors un message sera affiché «client existe déjà !!».

## 4)modification\_client()

Parmi les options que l'utilisateur a droit de choisir dans le menu 'Gestion des Clients' est : modifier client s'il veut modifier une ou plusieurs informations d'un client enregistré dans le fichier «client».

Pour rendre service à cet utilisateur on a créé la fonction **modification\_client()** dont on a appliqué le même principe de la fonction **modification\_voiture()** : D'abord on a déclaré deux pointeurs sur FILE , l'un ouvre le fichier «client» en mode lecture 'r', l'autre crée et ouvre un nouveau fichier «nouveau\_client» en mode écriture 'w'.

Après on demande d'utilisateur de saisir l'identifiant de la voiture qu'il souhaite modifier.

Au moment de la lecture du fichier «client» par la fonction **fscanf()** et la boucle **while**, on cherche le client dont l'id égal à celui inséré par l'utilisateur et on copie les infos dans le nouveau fichier «nouveau\_client», jusqu'à ce qu'on arrive à le trouver puis on change la valeur d'une variable déjà déclarée '**test**' de 0 à 1, et on demande d'utilisateur de choisir l'information qu'il souhaite modifier :1-nom,2-prénom,3-cin,4-adresse,5-telephone, après en lui demande de saisir la nouvelle information et on la stocke dans le nouveau fichier , ensuite on continue la copie des autres informations.

Enfin, on ferme les deux fichiers et on supprime l'ancien fichier « client » par :**remove("client.txt")** et on change le nom du nouveau fichier « nouveau\_client » par le nom « client » :**rename("nouveau\_client.txt","client.txt")** ;

Sans oublier de tester la valeur de la variable '**test**' s'elle garde la valeur 0, on affiche un message « Client introuvable !! » et on fait l'appel de la fonction **Fin2()** pour revenir au menu Gestion des Clients ou pour quitter l'application.

## 5)suppression\_client()

Pour supprimer un client du fichier «client», on a créé cette fonction **suppression\_client()** dont on déclare d'abord deux pointeurs sur FILE ,l'un ouvre le fichier «client» en mode lecture 'r' or l'autre crée un nouveau fichier «nouveau\_client» et l'ouvre en mode écriture 'w' ,similaire à ce qu'on a déjà fait dans la fonction qui précède.

Alors on passe à demander d'utilisateur d'entrer l'identifiant du client qu'il désire supprimer, après on introduit la fonction **fscanf()** et la boucle **while** pour lire les informations du fichier «client» : **while(fscanf(ancien,"%d...",&...) !=EOF) ;**

Si on trouve un id qui se diffère que celui inséré par l'utilisateur, on copie les informations de ce client dans le nouveau fichier «nouveau\_client»,or s'ils sont égaux on incrémente la valeur de la variable '**test**' qui était égale à 0,puis on affiche le message « Client bien supprimé », sans copier ses informations dans le nouveau fichier.

Enfin on ferme les deux fichiers et on supprime le fichier client et on renomme le nouveau fichier par le nom d'ancien fichier , puis on retourne pour faire le test sur la variable s'elle n'a pas changé de valeur ça veut dire que le client n'existe pas dans le fichier ,un message sera affiché pour le confirmer, et on appelle la fonction **Fin2()** à la fin.

# V-Location d'une voiture

Le but de notre agence , qu'on est en train de la développer par cette application, est la location des voitures. Donc ce menu 'Location d'une voiture' est le très important afin d'atteindre notre but, dans lequel on laisse le choix à l'utilisateur de choisir plusieurs options qu'on détaillera par suite à partir de leurs fonctions :

## 1)Saisie contrat()

Pour assurer la sécurité de notre service au sein de notre agence on demande de l'utilisateur de remplir un contrat qui sera stocker dans un fichier.

La fonction **Saisie\_contrat()** est créé afin de remplir ce fichier «contrat» qu'on a créé d'abord en déclarant un pointeur sur FILE et en ouvrant le fichier en mode 'w' :

**FILE \*f=fopen("contrat.txt","w") ;**

Puis on demande de l'utilisateur de saisir le nombre des contrats qu'il veut stocker dans le fichier «contrat», et on continue par lui demander de saisir les informations de chaque contrats suivant les sous variables de la structure

"ContratLocation" :numContrat/idVoiture/idClient/Date debut/Date fin/cout. Après on stocke les informations dans le fichier «contrat» en utilisant la fonction **fprintf(f,"%d...",&co.numContrat...);** Tout ça était fait au sein d'une boucle 'For' , sans oublier de fermer le fichier à la fin par la fonction **fclose(f)** .

## 2)visualisation contrat()

Pour s'assurer de la location d'une voiture par un client , l'utilisateur a besoin de vérifier le contrat déjà enregistrée dans le fichier «contrat», c'est pourquoi on a écrit la fonction **visualisation\_contrat()** dans laquelle on a ouvert premièrement le fichier «contrat» en mode lecture 'r' : **FILE \*f=fopen("contrat.txt","r") ;**

Puis on a demandé de l'utilisateur de saisir le numéro de contrat qu'il voudrait voir , ensuite on commence la lecture du fichier «contrat» par la fonction **fscanf()** et la boucle **while** : **while(fscanf(f,"%d...",&co.numContrat...) !=EOF)**, si on arrive au numéro de contrat qui est égal au numéro saisi par l'utilisateur on change la valeur de la variable 'test' de 0 à 1 et on affiche les informations contenus dans cette contrat par la fonction **printf()**.

Enfin, on ferme le fichier par la fonction **fclose(f)**, et on fait le test sur la variable 'test', si cette dernière n'a pas changée de valeur (càd test !=1) cela signifie que le contrat n'existe pas dans le fichier «contrat», un message sera affiché pour le confirmer. Et sans oublier de faire l'appel à la fonction **Fin3()** pour laisser le choix à l'utilisateur de revenir au menu 'Location d'une voiture' ou de quitter l'application.

### 3)louer\_voiture()

La fonction **louer\_voiture()** est très importante dans mon Mini-Projet , elle permet au client de louer le voiture qu'il veut à condition que cette dernière est disponible.

Alors dans cette fonction on a ouvert les deux fichiers «voiture» et «client» en mode lecture :**FILE \*p=fopen("client.txt","r") ; FILE \*ancien("voiture.txt","r") ;**  
et on a créé et ouvert un nouveau fichier en mode écriture :

**FILE\*nouveau("nouveau\_voiture","w") ;**

Puis on a passé à demander du client de saisir ses informations :idClient/nom/prénom d'après les sous variables de la structure "Client". Ensuite, on commence la lecture du fichier «client» par la boucle **while** et la fonction **fscanf()** , si on arrive à trouver un identifiant qui est égal à l'id saisi par l'utilisateur , on change la valeur de la variable '**test1**' de 0 à 1.

- Après on fait le test sur cette variable, si cette dernière n'a pas changé de valeur (càd test1 !=1) , cela signifie que le client n'est pas encore enregistré dans le fichier «client», et par suite un message sera affiché pour lui demander d'enregistrer d'abord ses informations dans l'option "Ajouter client" dans le menu "Gestion des Clients".
- Or si la variable '**test1**' a changé de valeur (càd test1=1) ,on passe à demander du client de saisir les informations de la voiture qu'il veut louer suivant le schéma de la structure "Voiture". Après on commence la lecture du fichier «voiture» par la fonction **fscanf()** et la boucle **while** : **while(fscanf(ancien,"%d....",&v.idVoiture....) !=EOF) ;**

Lors de notre lecture, si on arrive à un identifiant de voiture qui est égal à celui entré par l'utilisateur , on change la valeur d'une variable '**test2**' de 0 à 1.

Puis on passe à vérifier l'attribut 'En Location' de cette voiture par la fonction **strcmpi()**,  
▪si elle est de valeur 'oui' cela signifie que la voiture est déjà louée ,alors un message sera affiché pour l'approuver **«Excusez-nous voiture indisponible!!!»** en copiant les informations de la voiture dans le nouveau fichier «nouveau\_voiture» par la fonction **printf()**.

▪Sinon si la voiture n'est pas déjà en location càd (**strcmpi(v.EnLocation,"non")==0**), on change la valeur de la variable '**test3**' de 0 à 1,et puis on copie les informations de la voiture dans le nouveau fichier «nouveau\_voiture» avec le changement de la valeur de l'attribut 'EnLocation' de "Non" à "Oui".

Enfin, on termine la copie des informations des autres voitures dans le nouveau fichier, puis on ferme les fichiers par la fonction **fclose()** et on supprime l'ancien fichier «voiture» et on renomme le nouveau fichier «nouveau\_voiture» par le nom d'ancien fichier.

Sans oublier le test sur la variable 'test3' s'elle est égale à 1, on demande de l'utilisateur de remplir les informations du contrat suivant la structure 'ContratLocation' en ajoutant le contrat au fichier «contrat» qu'on a ouvert en mode d'ajout 'a' : **FILE**

**\*ptr=fopen("contrat .txt","a"),**

et aussi le test sur la variable '**test2**' si cette dernière n'a pas changé de valeur cela signifie que la voiture que le client souhaite louer n'est pas enregistrée dans le fichier alors on affiche un message pour lui informer et on appelle la fonction **Fin3()**.

## 4)retourner\_voiture()

Quand la durée de location de la voiture est fini pour notre client, ce dernier est obligé de rendre la voiture à l'agence en choisissant le troisième choix du menu 'Location d'une voiture'.

Et c'est là où le rôle de la fonction **retourner\_voiture()** est démontré. Dans cette fonction on a ouvert les deux fichiers «voiture» et «contrat» en mode lecture 'r'. Et on a créé et ouvert en même temps deux nouveau fichiers en mode écriture 'w' :

```
FILE *nouveauv=fopen("nouveau_voiture.txt","w") ;  
FILE *nouveauco=fopen("nouveau_contrat.txt","w") ;
```

Après on demande du client de saisir l'identifiant de la voiture qui a déjà louée. On commence la lecture du fichier «voiture» par la boucle **while** et la fonction **fscanf()** .

- Si on arrive à trouver un identifiant qui est égal à celui saisi par le client, on change la valeur de la variable '**test1**' de 0 à 1 pour montrer que la voiture existe dans le fichier et puis on copie les infos de la voiture dans le nouveau fichier «nouveau\_voiture» en changeant la valeur de l'attribut '**EnLocation**' de "Oui" à "Non".

- Sinon si on lit un identifiant autre que celui entré par l'utilisateur on suffit de copier les informations de la voiture telles qu'elles sont sans modification. A la fin de la lecture on affiche un message pour prouver le retour de la voiture «Voiture retournée !!» après on ferme les deux fichiers et on supprime le fichier «voiture» et on renomme le nouveau fichier par le nom d'ancien fichier.

Par la suite on passe à supprimer le contrat en lisant le fichier «contrat» par la fonction **fscanf()** et la boucle **while** ,

- si on trouve un identifiant de voiture différent de celui inséré par le client, on copie les informations de son contrat dans le nouveau fichier «nouveau\_contrat» .
- Sinon si les identifiants sont égaux on change la valeur de la variable '**test2**' de 0 à 1 pour assurer l'existence du contrat dans le fichier et puis on affiche le message «contrat supprimée !!».

Puis on ferme les deux fichiers et on supprime l'ancien fichier «contrat» et on renomme le nouveau fichier par le nom de l'ancien : **rename("nouveau\_contrat.txt","contrat.txt")** ;

Enfin on teste la valeur des deux variables '**test1**' et '**test2**' s'elles ne sont pas égales à 0 on affiche les deux messages «voiture n'existe pas» pour la variable 'test1' et «contrat n'existe pas» pour la variable 'test2' et on appelle la fonction **Fin3()** pour faire choisir le client de quitter l'application ou de revenir au menu 'Location d'une voiture'.

## 5) modifier\_contrat()

Afin de modifier une ou plusieurs informations contenues dans un contrat, on a créé la fonction `modifier_contrat()` dans laquelle on a ouvert d'abord le fichier «contrat» en mode lecture 'r' : `FILE *ancienco=fopen("contrat.txt","r") ;`

Puis on a créé et ouvert en même temps un nouveau fichier en mode écriture 'w' :  
`FILE *nouveauco=fopen("nouveau_contrat","w") ;`

Après on demande de l'utilisateur de saisir le numéro du contrat qu'il veut modifier, et on commence la lecture du fichier «contrat» par la boucle **while** et la fonction **fscanf()** en cherchant le contrat dont le numéro de rassemble à celui saisi par l'utilisateur et en même temps on copie les infos des autres contrats dans le nouveau fichier «nouveau\_contrat» .

Quand on trouve le numéro on change d'abord la valeur de la variable 'test1' de 0 à 1, puis on demande du client de choisir l'information qu'il veut modifier dans ce contrat :

\*\*\*\*\*

- Si le client choisit le premier choix (**choix=1**), on lui demande de saisir le nouveau identifiant de la voiture qu'il veut louée, et aussi de saisir l'ancien id de la voiture qui était déjà chez lui.

Ensuite on passe à chercher les ids dans le fichier «voiture» qu'on a ouvert en mode lecture 'r', si on trouve un id égal au nouveau id saisi par le client, on change la valeur de la variable 'test2' de 0 à 1 et on passe à vérifier la location de la voiture :

\*Si EnLocation='OUI', cela signifie que la voiture demandée est déjà louée, alors on affiche un message au client pour l'informer.

\*Sinon si EnLocation='NON', cela signifie que la voiture est disponible, on change la valeur de la variable 'test4', puis on copie les infos de la voiture dans le nouveau fichier «nouveau\_voiture» en changeant la valeur de l'attribut 'EnLocation' de 'NON' à 'OUI'.

\*Et si on arrive à trouver l'id qui est égal au ancien id, on copie les infos de la voiture dans le nouveau fichier «nouveau\_voiture» en changeant la valeur de l'attribut 'EnLocation' de 'OUI' à 'NON'.

Par suite, on termine la copie des infos des autres voitures dans le nouveau fichier «nouveau\_voiture», puis on ferme les deux fichiers par la fonction **fclose()**, et on supprime le fichier «voiture» et on renomme le nouveau fichier par le nom d'ancien :

`rename("nouveau_voiture.txt","voiture.txt") ;`

Après, on teste la variable 'test4' s'elle est égale à 1 on copie les infos du contrat dans le nouveau fichier en changeant l'id de la voiture par le nouveau id, sinon on copie les infos telles qu'elles sont.

Sans oublier enfin le test sur la variable 'test2', si cette dernière n'a pas changé de valeur on affiche un message « voiture introuvable ! ».

\*\*\*\*\*

- Si le client choisit le deuxième choix (**choix=2**), on lui demande de saisir son nouveau identifiant, puis on ouvre le fichier «client» en mode lecture 'r' et on commence la lecture des infos du fichier par la boucle **while** et la fonction **fscanf()**.

Si on trouve un id tel que celui saisi par le client, on change la valeur de la variable 'test3', et on copie les infos du contrat dans le nouveau fichier «nouveau\_contrat» en changeant l'id du client par le nouveau id.

Enfin, on teste la variable '**test3**' s'elle n'a pas changé de valeur cela signifie que le client n'est pas enregistré dans le fichier «client», un message sera affiché pour lui demander d'enregistrer ses infos d'après le menu 'Gestion des clients' et on copie les infos du contrat dans le nouveau fichier «nouveau\_contrat» telle qu'elles sont.

\*\*\*\*\*

- Or si le client choisit le troisième choix (**choix=3**), on lui demande de saisir la nouvelle date du début et on stocke le contrat dans le nouveau fichier «nouveau\_contrat» en changeant la valeur de l'attribut 'Date\_debut'.

- Or si (**choix=4**) , on demande du client d'entrer la nouvelle date de fin puis on copie les infos du contrat dans le nouveau fichier en changeant la date de fin.

- Sinon si (**choix=5**), on lui demande de saisir le nouveau cout de la voiture puis on copie les nouveau infos dans le nouveau fichier «nouveau\_contrat».

Mais si la valeur de la variable choix est supérieure à 5 on demande de l'utilisateur de faire entrer un chiffre entre 1 et 5.

Enfin, on ferme les deux fichiers «contrat» et «nouveau\_contrat» et on supprime le fichier «contrat» par la fonction : **remove()**, et on renomme le nouveau fichier par le nom d'ancien fichier : **rename("nouveau\_contrat.txt","contrat.txt")** ;

Sans oublier à la fin de faire le test sur la variable '**test1**' qu'on l' a changé de valeur quand on a trouvé le numéro du contrat qu'on veut modifier dans le fichier «contrat» , alors si la variable n'a pas changé de valeur cela signifie que le contrat n'existe pas dans le fichier « contrat», un message sera afficher pour l'approuver.

Et enfin on appelle la fonction **Fin3()** pour laisser le choix à l'utilisateur de quitter l'application ou de revenir au menu 'Location d'une voiture'.