UNIVERSITE IBN ZOHR AGADIR

FACULTER DE SCIENCE

DEPARTEMENT INFORMIQUE

Titre de Mini Projet en Langage C :

Gestion de location de voiture

Projet réalisé par :

Khadija ABDELWASSAA

Gmail :

Khadijaabdelwassaa20@gmail.com

Projet encadré par :

Fouad El Ouafdi

SOMMAIRE

Représentation :

La location des voitures est devenue un secteur en pleine expansion, dont la compétitivité augmente jours après jours, donc pour améliorer ces services une agence de location de voiture décide d'informatiser sont système de gestion à travers la réalisation d'une base de données regroupant tous les informations concernant ses clients, ses contrats, ces voitures, afin de facilité la tâche aussi bien pour le client que pour son personnelles.

Les problématiques ☹ et leurs solutions ☺ :

Dans ce projet je trouver beaucoup des facultatifs :

- ❖ Travaille avec GTK :

  Pour crées une seule fenêtre j'ai codé par 5 lignes ou maximum 😐 c'est pour ça j'ai travaillé par **glade** 💣 car est plus simple d'utiliser et n'est pas besoin de 5 ligne juste une seule ligne de code.

- ❖ Travaille avec les fichiers et GTK :

  Problème est comment afficher par exemple une List des informations dans une fenêtre et ou même temps d'enregistrer cette liste dans un fichier, et pour ça je propose d'utilise **le consule** 🕯🕯 (terminal).

Introduction à GTK

Définition :

GTK (*The **G**IMP **T**ool**k**it*, anciennement **GTK+**)

est un ensemble de bibliothèques logicielles, c'est-à-dire un ensemble de fonctions permettant de réaliser des interfaces graphiques. Cette bibliothèque a été développée originellement pour les besoins du logiciel de traitement d'images GIMP. GTK+ est maintenant utilisé dans de nombreux projets, dont les environnements de bureau GNOME, Xfce, Lxde et ROX.

GTK est une bibliothèque développée en C, c'est-à-dire que Gtk met à disposition des structures et des méthodes que l'on peut intégrer à un programme C. Parmi ces structures, on trouve des objets graphiques, tels que la fenêtre, le bouton, le label, etc.

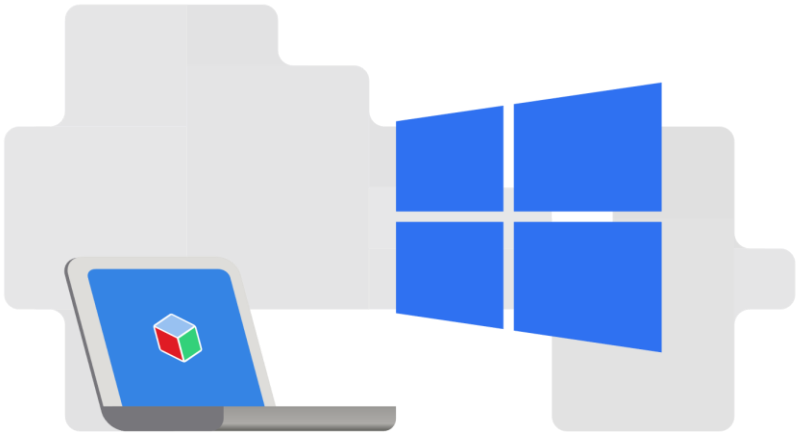
Version de GTK :

- GTK+2
- GTK+3
- GTK+4

Dans ce projet en travail avec GTK+3

```
#include<gtk/gtk.h>
```

Installer GTK+ sous Windows:



Installer GTK+ sous windows on utilise packages **MSYS2**.

- Téléchargez le programme d'installation **MSYS2** qui correspond à votre plate-forme et suivez les instructions d'installation dans ce site： https://www.msys2.org/
- Installer GTK3, Ouvrez un shell MSYS2 et exécutez ces commandes par ordre ：

Steep1: pacman -S mingw-w64-x86_64-gtk3

Steep2: pacman -S mingw-w64-x86_64-pkg-config

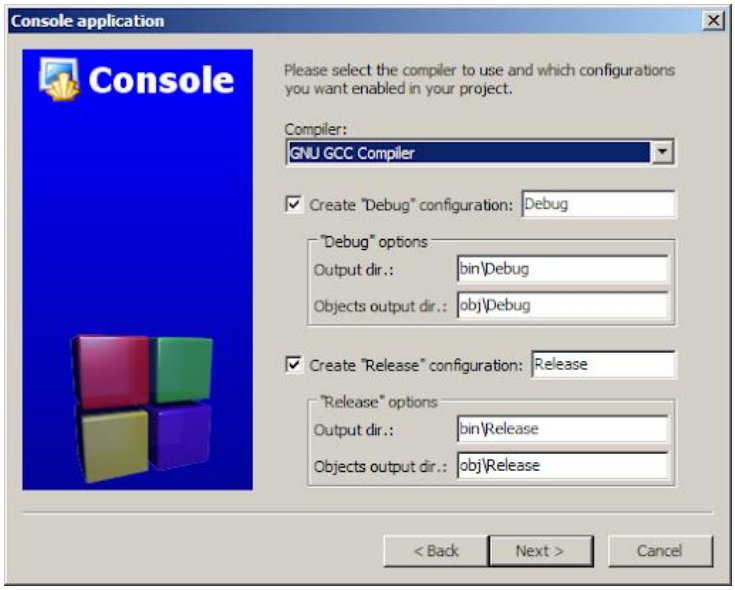
Steep3: pacman -S mingw-w64-x86_64-glade

Steep4: pacman -S mingw-w64-x86_64-toolchain base-devel

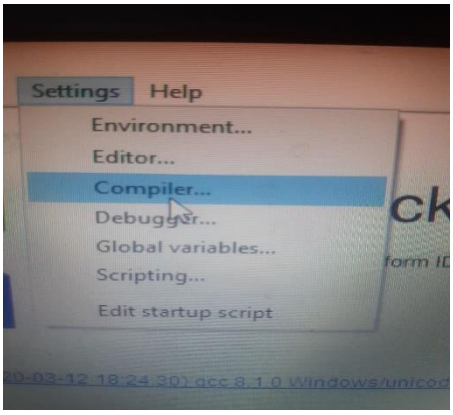
GTK sous Code block:

➢ ***Crier un new project***

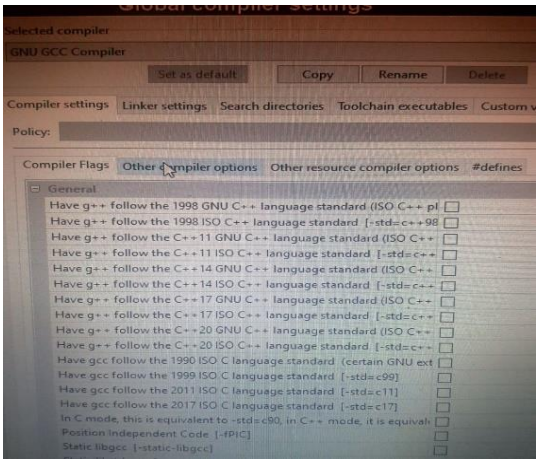
1) Open code block.
2) Click sur new project.
3) Click sur console application.
4) 

5) 

- ***<u>Aller dans les paramètres et clique sur compiler.</u>***



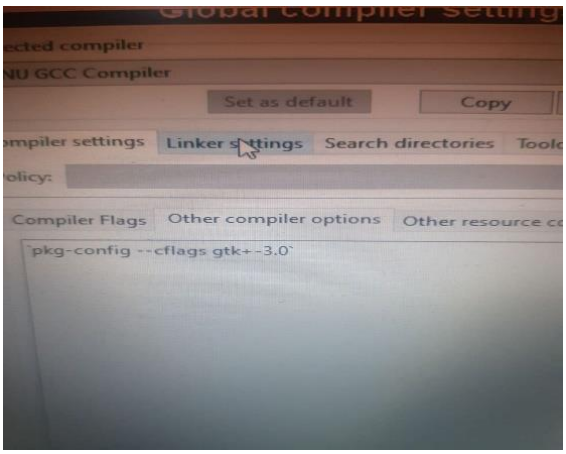
1) Click sur « Other compiler options »



2) Copier le command suivant :

`pkg-config --cflags gtk+-3.0`

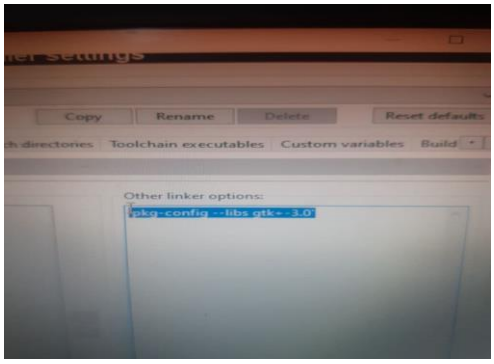
3) Click sur « linker settings »



4) Copier le command suivant :

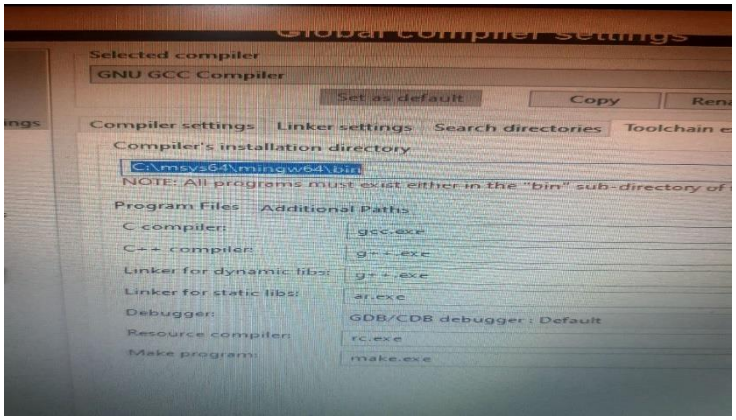
`pkg-config --libs gtk+-3.0`

- Dans la fenêtre « Other compiler options »



5) Click sur « toolchain executables » :

Copier toutes les informations dans l'image En dessous.



6) Click « Ok »

Interface graphique avec glade :

Introduction

Glade est un outil interactif de conception d'interface graphique GTK+. Il prend en charge toute la partie de gestion/génération de l'interface pour permettre au développeur de se concentrer sur le code « utile ».

Glade enregistre les interfaces graphiques en générant des fichiers XML.

Initialement la bibliothèque **libglade** permettait de lire ces fichiers dynamiquement (c'est-à-dire à l'exécution de l'application). **Gt Builder** (inclus dans GTK+) remplace dorénavant libglade dans l'environnement de bureau GNOME .

Dictionaries de donnée:

1) Variable globale d'interface graphique :

```
static GtkBuilder *b;

static GtkWidget *windowM;

static GtkWidget *window1;

static GtkWidget *window3;

static GtkWidget *window5;

static GtkWidget *window4;
```

2) Structure Menu des buttons:

Contient une List des buttons, on utilise pour chaque fonction pour éviter répétition des variables et le blocage de mémoire.

```
typedef struct
{
    GtkWidget *button1;
    GtkWidget *button2;
    GtkWidget *button3;
    GtkWidget *button4;
    GtkWidget *button5;
    GtkWidget *button6;

}menu;
```

les functions interface graphique:

void Menu principal ();

Permet d'afficher l'interface de Menu principal qui contient des buttons.

Code source :

```
void Menu_principal()
{
      menu M;
      b=gtk_builder_new_from_file("c.glade");
      window1=GTK_WIDGET(gtk_builder_get_object(b,"w1"));
      M.button1=GTK_WIDGET(gtk_builder_get_object(b,"button1_w2"));
      M.button3=GTK_WIDGET(gtk_builder_get_object(b,"button3_w2"));
      M.button2=GTK_WIDGET(gtk_builder_get_object(b,"button2_w2"));
      M.button4=GTK_WIDGET(gtk_builder_get_object(b,"button4_w2"));

      //connect chaque button par leur fonction

      g_signal_connect(M.button1 , "clicked" ,G_CALLBACK(Gestion_location), NULL);
      g_signal_connect(M.button2, "clicked" ,G_CALLBACK(Gestion_voiture), NULL);
      g_signal_connect(M.button3, "clicked" ,G_CALLBACK(Gestion_client), NULL);
      g_signal_connect(M.button4 , "clicked" ,G_CALLBACK(gtk_main_quit), NULL);

      //destroy la windows pour chaque click de buttom

      g_signal_connect(M.button1 , "clicked" ,G_CALLBACK(close1), NULL);
       g_signal_connect(M.button2 , "clicked" ,G_CALLBACK(close1), NULL);
      g_signal_connect(M.button3 , "clicked" ,G_CALLBACK(close1), NULL);
      g_signal_connect(M.button3 , "clicked" ,G_CALLBACK(close1), NULL);

      gtk_widget_show_all(window1);
}
```

#menu M : nouveau type de structure « menu ».

 # b : open le fichier « c.glade » qui contient des fenêtre.

 # window1 : appelle de fenêtre dans le fichier « c.glade » qui a identité « w1 ».

 # M.button1 : appelle le buttons dans le fichier « c.glade » qui a identité « button1_w2 » de titre «Location » .

 # M.button2 : appelle le buttons dans le fichier « c.glade » qui a identité « button2_w2 » de titre «Gestion voiture » .

 # M.button3 : appelle le buttons dans le fichier « c.glade » qui a identité « button3_w2 » de titre «Gestion client » .

 # M.button4 : appelle le buttons dans le fichier « c.glade » qui a identité « button4_w2 » de titre «Quitter» .

 # g_signal_connect(M.button1 ,"clicked" ,G_CALLBACK(Gestion_location), NULL):

Signal connecter entre M.button1 et fonction « Gestion_location » c'est-à-dire tant que on clique sur la button de location le signal va connecté avec fonction « Gestion_location »

 # g_signal_connect(M.button2, "clicked" ,G_CALLBACK(Gestion_voiture), NULL);

Signal connecter entre M.button2 et fonction « Gestion_voiture » c'est-à-dire tant que on clique sur la button de location le signal va connecté avec fonction « Gestion_voiture »

#g_signal_connect (M.button3, "clicked" ,G_CALLBACK(Gestion_client), NULL);

Signal connecter entre M.button3 et fonction « Gestion_client » c'est-à-dire tant que on clique sur la button de location le signal va connecté avec fonction « Gestion_client »

#g_signal_connect (M.button4 , "clicked" ,G_CALLBACK(gtk_main_quit), NULL);

Ce signal va Quitter le programme si on clique sur le Button « Quitter » de la fenêtre « Menu_Principal »

- **gtk_main_quit :** fonction prédéfinie permet de quitter le programme.

#g_signal_connect (M.button(1,2,3,4),"clicked", G_CALLBACK(close1), NULL);

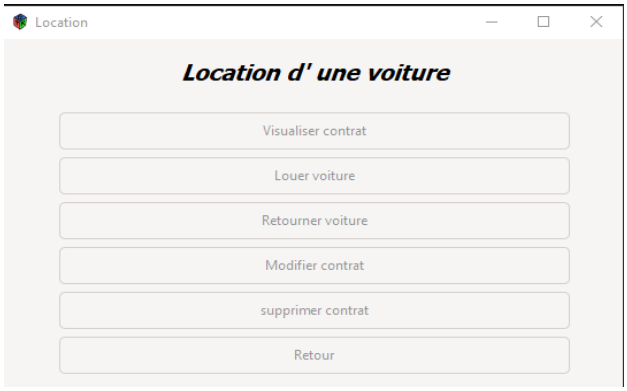
Signal connecter entre tous les de List M.button(1,2 ;3,4) et fonction « close1» c'est-à-dire tant que on clique sur chaque button de fenêtre menu principal le signal va destroy ce fenêtre .

```
void close1() //destroy Menu principal
{
    gtk_widget_destroy(window1);
}
```

- **Conclusion :**
  En général, l'interface Menu principal connecté avec **les sous interface** (location ; gestion voiture ; etc…).

- Void Gestion_location () ;

Permet d'afficher l'interface location quand on clique sur la Button de location dans l'interface Menu principal.



- Code source :

```
void Gestion_location()
{
      b=gtk_builder_new_from_file("c.glade");
      window5=GTK_WIDGET(gtk_builder_get_object(b,"w2"));
      M.button1=GTK_WIDGET(gtk_builder_get_object(b,"Vc"));
      M.button2=GTK_WIDGET(gtk_builder_get_object(b,"Lc"));
      M.button3=GTK_WIDGET(gtk_builder_get_object(b,"Rc"));
      M.button4=GTK_WIDGET(gtk_builder_get_object(b,"Mc"));
      M.button5=GTK_WIDGET(gtk_builder_get_object(b,"Sc"));
      M.button6=GTK_WIDGET(gtk_builder_get_object(b,"Rw"));
      g_signal_connect(M.button1 , "clicked" ,G_CALLBACK(visualiser_contrat), NULL);
      g_signal_connect(M.button2 , "clicked" ,G_CALLBACK(louer_voiture), NULL);
      g_signal_connect(M.button3 , "clicked" ,G_CALLBACK(Return_voiture), NULL);
      g_signal_connect(M.button4 , "clicked" ,G_CALLBACK(Modifier_contrat), NULL);
      g_signal_connect(M.button5 , "clicked" ,G_CALLBACK(supprimer_contrat), NULL);
      g_signal_connect(M.button6 , "clicked" ,G_CALLBACK(Retourlocation), NULL);
       gtk_widget_show_all(window5);
}
```

 # b : open le fichier « c.glade » qui contient des fenêtre.

 # window5 : appelle de fenêtre dans le fichier « c.glade » qui a identité « w2» .

 # M.button1 : appelle le buttons dans le fichier « c.glade » qui a identité « vc » de titre «visauliser contrat » .

 # M.button2 : appelle le buttons dans le fichier « c.glade » qui a identité « lc » de titre «louer voiture » .

 # M.button3 : appelle le buttons dans le fichier « c.glade » qui a identité « Rc» de titre «retourner voiture » .

 # M.button4 : appelle le buttons dans le fichier « c.glade » qui a identité « Mc» de titre «Modifier contrat» .

 # M.button5 : appelle le buttons dans le fichier « c.glade » qui a identité «Sc» de titre «supprimer le contrat» .

 # M.button6 : appelle le buttons dans le fichier « c.glade » qui a identité « Rw» de titre «retour» .

 # g_signal_connect(M.button1 , "clicked" ,G_CALLBACK(visualiser_contrat), NULL):

Signal connecter entre M.button1 et fonction « visualiser_contrat» c'est-à-dire tant que on clique sur la Button de visualiser contrat le signal va connecté avec fonction « visualiser_contrat».

#g_signal_connect(M.button2 , "clicked" ,G_CALLBACK(louer_voiture), NULL):

Signal connecter entre M.button2 et fonction « louer_voiture» c'est-à-dire tant que on clique sur la Button de louer voiture le signal va connecté avec fonction « louer_voiture».

#g_signal_connect(M.button3 , "clicked" ,G_CALLBACK(Return_voiture), NULL) :

Signal connecter entre M.button3 et fonction «returner_voiture» c'est-à-dire tant que on clique sur la Button de returner voiture le signal va connecté avec fonction « returner_voiture».

#g_signal_connect(M.button4 , "clicked" ,G_CALLBACK(Modifier_contrat), NULL):

Signal connecter entre M.button4 et fonction «Modifier_contrat» c'est-à-dire tant que on clique sur la Button de Modifier contrat le signal va connecté avec fonction « Modifier_contrat».

#g_signal_connect(M.button5 , "clicked" ,G_CALLBACK(supprimer_contrat), NULL):

Signal connecter entre M.button5 et fonction «supprimer_contrat» c'est-à-dire tant que on clique sur la Button de supprimer contrat le signal va connecté avec fonction «supprimer_contrat ».

g_signal_connect(M.button6 , "clicked" ,G_CALLBACK(Retourlocation), NULL):

Signal connecter entre M.button6 et fonction «Retourlocation» c'est-à-dire tant que on clique sur la Button de Retour location le signal va connecté avec fonction «Retourlocation».

→Void Retourlocation();

Fonction permet de retour a l'interface Menu principal avec la destroy de l'interface location de voiture .

```
void Retourlocation()
{
        Menu_principal();
        gtk_widget_destroy(window5);

}
```

- **Conclusion :**

En général, l'interface location de voiture connecté avec **les sous fonction code en langage C.**

- Void Gestion_voiture () ;

Permet d'afficher l'interface Gestion de voiture quand on clique sur la Button de gestion de voiture dans l'interface Menu principal.

- Code source :

```
void Gestion_voiture()
{
      b=gtk_builder_new_from_file("c.glade");
      window3=GTK_WIDGET(gtk_builder_get_object(b,"w3"));
      M.button1=GTK_WIDGET(gtk_builder_get_object(b,"Lv"));
      M.button2=GTK_WIDGET(gtk_builder_get_object(b,"Av"));
      M.button3=GTK_WIDGET(gtk_builder_get_object(b,"Mv"));
      M.button4=GTK_WIDGET(gtk_builder_get_object(b,"Sv"));
      M.button5=GTK_WIDGET(gtk_builder_get_object(b,"Rv"));
      g_signal_connect(M.button1 , "clicked" ,G_CALLBACK(list_voiture), NULL);
      g_signal_connect(M.button2 , "clicked" ,G_CALLBACK(ajoute_voiture), NULL);
      g_signal_connect(M.button3 , "clicked" ,G_CALLBACK(modifier_voiture), NULL);
      g_signal_connect(M.button4 , "clicked" ,G_CALLBACK(suprimer_voiture), NULL);
      g_signal_connect(M.button5 , "clicked" ,G_CALLBACK(Retourvoiture), NULL);
     gtk_widget_show_all(window3);
}
```

 # b : open le fichier « c.glade » qui contient des fenêtre.

 # window3 : appelle de fenêtre dans le fichier « c.glade » qui a identité « w3» .

 # M.button1 : appelle le buttons dans le fichier « c.glade » qui a identité « Lv » de titre «List de voiture» .

 # M.button2 : appelle le buttons dans le fichier « c.glade » qui a identité « av » de titre «ajouter voiture» .

 # M.button3 : appelle le buttons dans le fichier « c.glade » qui a identité « mv» de titre «modifier voiture » .

 # M.button4 : appelle le buttons dans le fichier « c.glade » qui a identité « sv» de titre «supprimer voiture» .

 # M.button5 : appelle le buttons dans le fichier « c.glade » qui a identité « Rv» de titre «retour» .

#g_signal_connect(M.button1 , "clicked" ,G_CALLBACK(list_voiture), NULL);

Signal connecter entre M.button1 et fonction « list_voiture»c'est-à-dire tant que on clique sur la Button de list voiture le signal va connecté avec fonction « list_voiture ».

#g_signal_connect(M.button2 , "clicked" ,G_CALLBACK(ajoute_voiture), NULL);

Signal connecter entre M.button2 et fonction «ajoute_voiture» c'est-à-dire tant que on clique sur la Button de ajoute_voiture le signal va connecté avec fonction « ajoute_voiture».

#g_signal_connect(M.button3 , "clicked" ,G_CALLBACK(modifier_voiture), NULL);

Signal connecter entre M.button3 et fonction «modifier_voiture» c'est-à-dire tant que on clique sur la Button de modifier voiture le signal va connecté avec fonction « modifier_voiture».

#g_signal_connect(M.button4 , "clicked" ,G_CALLBACK(suprimer_voiture), NULL);

Signal connecter entre M.button4 et fonction «suprimer_voiture» c'est-à-dire tant que on clique sur la Button de supprimer voiture le signal va connecté avec fonction « suprimer_voiture».

 # g_signal_connect(M.button5 , "clicked" ,G_CALLBACK(Retourvoiture), NULL);

Signal connecter entre M.button6 et fonction «Retourvoiture» c'est-à-dire tant que on clique sur la Button de Retour voiture le signal va connecté avec fonction «Retourvoiture».

→Void Retourvoiture () ;

Fonction permet de retour à l'interface Menu principal avec la destroy de l'interface Gestion de voiture.

```
void Retourvoiture()
{
	Menu_principal();
	gtk_widget_destroy(window3);

}
```

- **Conclusion**

En général, l'interface Gestion de voiture connecté avec **les sous fonction code en langage C.**

- void Gestion_client();

Permet d'afficher l'interface Gestion de client quand on clique sur la Button de gestion de client dans l'interface Menu principal.

- Code source :

```
void Gestion_client()
{

      b=gtk_builder_new_from_file("c.glade");
      window4=GTK_WIDGET(gtk_builder_get_object(b,"w4"));
      M.button1=GTK_WIDGET(gtk_builder_get_object(b,"Lcl"));
      M.button2=GTK_WIDGET(gtk_builder_get_object(b,"Acl"));
      M.button3=GTK_WIDGET(gtk_builder_get_object(b,"Mcl"));
      M.button4=GTK_WIDGET(gtk_builder_get_object(b,"Scl"));
      M.button5=GTK_WIDGET(gtk_builder_get_object(b,"Rcl"));
      g_signal_connect(M.button1 , "clicked" ,G_CALLBACK(list_Clients), NULL);
      g_signal_connect(M.button2 , "clicked" ,G_CALLBACK(ajoute_Client), NULL);
      g_signal_connect(M.button3 , "clicked" ,G_CALLBACK(modifier_Client), NULL);
      g_signal_connect(M.button4 , "clicked" ,G_CALLBACK(suprimer_client), NULL);
      g_signal_connect(M.button5 , "clicked" ,G_CALLBACK(Retourclient), NULL);
     gtk_widget_show_all(window4);
}
```

Remarque :

Gestion de client travaille comme Gestion de voiture dans le même concept.

Fonction main () :

Fonction main () est un ’interface représentative.

- **Code source :**

✓ gtk_init (&argc, &argv)

Appelez cette fonction avant d'appeler n'importe quelle autre fonction dans vos application GUI. Elle initialisera tout ce qui est nécessaire pour faire fonctionner la boîte à outils et analyser certaines options standards de ligne de commande. argc et argv are ajustés en conséquence ainsi votre propre code ne verra jamais ces arguments standards.

- ✓ **b** : open le fichier « c.glade » qui contient des fenêtre.

- ✓ **windowM** : appelle de fenêtre dans le fichier « c.glade » qui a identité « wM ».

- ✓ **M.buttonA** : appelle le buttons dans le fichier « c.glade » qui a identité «button2» de titre «Menu principal » .

g_signal_connect(buttonA,"clicked",G_CALLBACK(Menu_principal), NULL):

Signal connecter entre M.buttonA et fonction «Menu_principal» c'est-à-dire tant que on clique sur la Button de menu principal le signal va connecté avec cette fonction et après va afficher la fenêtre Menu principal.

g_signal_connect(buttonA , "clicked" ,G_CALLBACK(close), NULL);

Signal connecter entre buttonA et fonction « close» c'est-à-dire tant que on clique sur buttonA de fenêtre « miniprojet » le signal va destroy ce fenêtre .

```
void close()//destroy premier window
{
    gtk_widget_destroy(windowM);
}
```

Code sources en langage C :

Introduction:

Le C est un langage de programmation de bas niveau très populaire, crée dans les années 1970 par D.Ritchie et B.W.Kernighan. Il est portable, libre, faiblement typé (peu de types de variables différents : son fonctionnement est donc proche de l'ordinateur (gain en rapidité), mais un brin plus difficile à manipuler pour le programmeur). Le C n'est sans doute pas le langage le plus facile à apprendre (notamment à cause de l'adoption du concept parfois un peu obscure des pointeurs), ni le plus récent, mais ses qualités font de lui un langage incontournable en matière de programmation.

Dictionnaires de donnée :

- Structure contrat location enregistre dans un fichier « contrat .txt ».

```
typedef struct
{
        float num_contrat ;
        int id_Voiture;
        int id_client;
        date_debut debut;
        date_fin fin;
        int count;

}contrat;
contrat cl;
```

Avec date_debut et data_fin sont des structures comme suivant →

```
typedef struct
{
        int jj_debut;
        int mm_debut;
        int aa_debut;

}date_debut;
typedef struct
{
        int jj_fin;
        int mm_fin;
        int aa_fin;

}date_fin;
```

- **Structure voiture enregistre dans un fichier « voiture .txt ».**

```
typedef struct
{

        int id_voiture;
        char marque[15];
        char nom_voiture[15];
        char couleur[7];
        int nbplace;
        int prix_jour;
        char enlocation[4];
}voiture;
voiture V;
```

Avec définition d'un nouveaux type « V » de voiture, comme une variable globale.

- **Structure client enregistre dans le fichier « client.txt »**

```
typedef struct client
{
      int id_client;
      char nom[20];
      char prenom[20];
      int cin;
      char adresse[15];
      int telephone;
}client;
client Client;
```

Avec définition d'un nouveaux type « Client » de client, comme une variable globale.

Les fonctions en langage c :

❖ Deus fonction plus utiliser dans le programme :

Le premier fait la recherche dans le fichier « voiture.txt » de id voiture s'il existe retourner 1, sinon retourner -1.

```
int recherche_voiture(int Vrech)
{
      FILE *F;
      F=fopen("voiture.txt","r");
      fflush(stdin);
      do
      {
        fscanf(F,"%d,%s,%s,%s,%d,%d,%s\n" ,&V.id_voiture,&V.marque,&V.nom_voiture,&V.couleur,&V.nbplace,&V.prix_jour,&V.enlocation);
        if(V.id_voiture == Vrech)
        {
              fclose(F);
              return 1;

        }
      }while(!feof(F));
        fclose(F);
        return -1;
}
```

Le deuxième fait la recherche dans le fichier « client.txt » de id client s'il existe retourner 1, sino retourner -1.

```
int recherche_Client(int Crech)
{
      FILE *F;
      F=fopen("Client.txt","r");
      fflush(stdin);
      do
      {
        fscanf(F,"%d, %s, %s, %d, %s, %d \n" ,&Client.id_client,&Client.nom,&Client.prenom,&Client.cin,&Client.adresse,&Client.telephone
        if(Client.id_client == Crech)
        {
              fclose(F);
              return 1;

        }
      }while(feof(F));
        fclose(F);
        return -1;
}
```

→Gestion de location

❶Visualiser le contrat :

Permet de visualiser les informations dans un contrat à partir de son numéro qui a enregistré dans le champ num_contrat sur structure.

```
void visualiser_contrat()
{
      gtk_widget_destroy(window5);
      system("cls");
    float num_contrat;
    contrat Contrat;
    printf("Entrez le numero de contrat que tu veux l'afficher: \n");
    scanf("%f", &num_contrat);
    FILE* fichier;
    fichier=fopen("ContratsLocations.txt", "r");
    do {
      fscanf(fichier, "%f,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,\n",
                  &Contrat.num_contrat, &Contrat.id_Voiture, &Contrat.id_client,
                  &Contrat.debut.jj_debut, &Contrat.debut.mm_debut, &Contrat.debut.aa_debut,
                  &Contrat.fin.jj_fin, &Contrat.fin.mm_fin, &Contrat.fin.aa_fin, &Contrat.count);
      if(num_contrat==Contrat.num_contrat) {
            printf("\t\tNumero de contrat : %f",Contrat.num_contrat);
            printf("\t\tid de voiture  : %d",Contrat.id_Voiture);
            printf("\t\tid de client : %d",Contrat.id_client);
            printf("\t\t Le debut de location : %d-%d-%d", Contrat.debut.jj_debut, Contrat.debut.mm_debut, Contrat.debut.aa_debut);
            printf("\t\t La fin de location : %d-%d%d", Contrat.debut.aa_debut, Contrat.fin.mm_fin, Contrat.fin.aa_fin);
            printf("\t\t Le cout: %d dh", Contrat.count);
            break;
        }
      }while(!feof(fichier));

      fclose(fichier);
}
```

gtk_widget_destroy(window5)→destroy la fenêtre “gestion de location”.

System(cls)→clears le console.

On demande à l’utilisateur d’entrer son numéro de contrat et ouvrir le ficher « contratlocation.txt »et on compare avec numéro saisie avec numéro par un boucle qui lire le fichier, s il est existé donc va afficher dans le consol contrat de c’utilisateur on faire ça par fonction **Printf** et on fait **break** pour sortir, s’il termine va « gestion location ».

❷Louer voiture 📄🖉 :

gtk_widget_destroy(window5)→destroy la fenêtre “gestion de location”.

System(cls)→clears le console.

Permet de louer une voiture mais **en premier partie** il faut vérifier que client est existé déjà ou pas si non en demande ou client de saisie des informations à partir de **la fenêtre Gestion de client** s’il choisit « 1 » et après retourner 0 sinon « 0 » affiche la fenêtre **Menu principal** après sortir.

```
int louer_voiture()
{
     gtk_widget_destroy(window5);
      system("cls");
    int idc,idv;
    int trouver_client=0;
    int trouver_voiture=0;
    voiture V;
    contrat Contrat;
    printf("Donnez id de client : ");
    scanf("%d",&idc);
    trouver_client=recherche_Client(idc);
    if(trouver_client==-1) { /* if client n'est exist pas en fait l appelle de gestion_client()*/
       int i;
        printf("\n\n\t\tERROR votre id n'est existe pas\n");
        printf("enregistrer des propres information a travers le menu de gestion client , 1|0");
        scanf("%d",&i);
        switch(i)
        {
            case 1:
             Gestion_client();
             return 0;
            case 0:
                  Menu_principal();
                 return 0;
            default: printf("vous voulez entrer 1|0");
        }
    }
```

En deuxième partie s’il client existe on vérifier si la voiture existe, demande d’abord ou client de saisie le id voiture qui veut à louer s’il n’existe pas afficher une erreur.

```
/* if client dejat exist */

printf("donnez id de voiture : ");
scanf("%d",idv);
trouver_voiture=recherche_voiture(idv);
if(trouver_voiture==-1) {  /* if la voiture n'est exist pas*/
    printf("\n\n\t\tERROR voiture n'est existe pas\a");
    getchar();
}
```

En troisième partie s'il le voiture existe on si le champ Enlocation de structure voiture contient « oui » ou « non », c'est oui donc ne peut louer la voiture. C'est non allé remplir le contrat avec le changement le contenu de champ Enlocation par « oui », s'il termine va « gestion location ».

```
  /* if la voiture exist*/

  else {  /* en vus if la voiture en location ou no*/
      FILE *F;
      FILE *N;
      F=fopen("voiture.txt","r");
      N=fopen("nouveu_voiture.txt", "w"); /* ce fichier va copier le data de fichier "voiture.txt" pour changer la valeur de enlovation */
      fflush(stdin);
      /* cherche de voiture de fichier "voiture.txt" */
      do {
          fscanf(F,"%d,%s,%s,%s,%d,%d,%s/n" ,V.id_voiture,V.marque,V.nom_voiture,V.couleur,V.nbplace,V.prix_jour,V.enlocation);

          /* if a voiture exist */
          if(V.id_voiture == idv) {
              if( strcmp(V.enlocation,"oui")==0) { /* if voiture deja louer*/
                  printf("\n\n\t\tvoiture deja louer\a");
                  getchar();
                  fclose(F);  /* fermeture les fichier */
                  fclose(N);
                  remove("nouveu_voiture.txt");  /* supprimer le fichier */
                  return 1; /* sortier car on ne peut contuner avec un voiture deja louer */
              }
              else {  /* if la voiture possible de louer*/
                  printf("\t\t\t\tclient,saiaie les information necessaire suivant: \n ");
                  /* ecrire le contrat*/
                  /* id voiture et id de client deja exist */
                  printf("numero du contrat: ");
                  scanf("%f", &Contrat.num_contrat);
                  printf("\n\t donnez la date debut de location (jj mm aa): ");
                  scanf("%d %d %d", &Contrat.debut.jj_debut, &Contrat.debut.mm_debut,&Contrat.debut.aa_debut);
                  printf("\t\t\tentrez date de fin (jj mm aa): ");
                  scanf("%d %d %d", &Contrat.fin.jj_fin, &Contrat.fin.mm_fin, &Contrat.fin.aa_fin);
                  printf("\t\t Determinez le count: ");
                  scanf("%d", &Contrat.count);
                  /*On ajout cette contrat dans le fichier "contatlocation"*/
                  FILE* fichier;
                  fichier = fopen("ContratsLocations.txt", "a");
                      fprintf(fichier, "%f, %d, %d, %d, %d, %d, %d, %d, %d, %d, \n",
                              Contrat.num_contrat, idv, idc,
                              Contrat.debut.jj_debut, Contrat.debut.mm_debut, Contrat.debut.aa_debut,
                              Contrat.fin.jj_fin, Contrat.fin.mm_fin, Contrat.fin.aa_fin, Contrat.count
                      );
                  fclose(fichier);
                  /* change la valeur de enlocation*/
                  strcpy(V.enlocation,"oui");
                  /* copier dans un notre fichier*/
                  fprintf(N, "%d,%s,%s,%s,%d,%d,%s/n" ,V.id_voiture,V.marque,V.nom_voiture,V.couleur,V.nbplace,V.prix_jour,V.enlocation);
                  continue; /* pour eviter repetion de fprintf */
              }
          }
          /* copier  "voiture" dans "nouveu_voiture"*/
          fprintf(N, "%d,%s,%s,%s,%d,%d,%s/n" ,V.id_voiture,V.marque,V.nom_voiture,V.couleur,V.nbplace,V.prix_jour,V.enlocation);
      }while(!feof(F));
      fclose(F);
      fclose(N);
      remove("voiture.txt");
      rename("nouveu_voiture.txt", "voiture.txt");

  }
}
```

❸Retourner voiture 🔑:

Permet de retourner une voiture déjà louer par un client avec un changement de contenu Enlocation « oui » par « non ».

gtk_widget_destroy(window5)→destroy la fenêtre “gestion de location”.

System(cls)→clears le console.

En premier, ouvrir le fichier on mode lecture et tester s’il le voiture existe si « oui » va copier les donner de fichier « voiture.txt » dans le fichier « nouveauvoiture.txt » avec un changement de Enlocation par « non ».

```c
void Return_voiture() {
     gtk_widget_destroy(window5);
     system("cls");
    int Vrech;
    contrat Contrat;
    FILE *F;
    FILE *N;

    printf("Donneez id de la voiture: ");
    scanf("%d", &Vrech);

    F=fopen("voiture.txt", "r");
    N=fopen("nouveu_voiture.txt", "w");
    do {
        fscanf(F,"%d,%s,%s,%s,%d,%d,%s/n" , &V.id_voiture, V.marque, V.nom_voiture, V.couleur, &V.nbplace, &V.prix_jour, V.enlocation);

        /* if la voiture exist */
        if(V.id_voiture == Vrech) {
            strcpy(V.enlocation,"non");
            /* copier les donner de fichier voiture dans le fichier "nouveu_voiture.txt" en chande  enlocation = "non" */
            fprintf(N, "%d,%s,%s,%s,%d,%d,%s/n" ,V.id_voiture,V.marque,V.nom_voiture,V.couleur,V.nbplace,V.prix_jour,V.enlocation);
            continue;
        }
        /* copier les donners de fichier voiture dans le fichier "nouveu_voiture.txt"*/
        fprintf(N, "%d,%s,%s,%s,%d,%d,%s/n" ,V.id_voiture,V.marque,V.nom_voiture,V.couleur,V.nbplace,V.prix_jour,V.enlocation);
    }while(!feof(F));
```

En deuxième, supprimer le contrat, s’il termine va « gestion location ».

```c
    fclose(F);
    fclose(N);
    remove("voiture.txt");
    rename("nouveu_voiture.txt", "voiture.txt");

    /* supprimer le contrat */
    /* recherche l idvoiure de contrat si elle est meme qui le client saisie */
  /* cherche dans le voiture*/
    F=fopen("ContratsLocations.txt", "r");
    N=fopen("nouveu_Contrat.txt", "w"); /* new fichier */
    do {
        fscanf(F, "%f,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,\n",
            &Contrat.num_contrat, &Contrat.id_Voiture, &Contrat.id_client,
            &Contrat.debut.jj_debut, &Contrat.debut.mm_debut, &Contrat.debut.aa_debut,
            &Contrat.fin.jj_fin, &Contrat.fin.mm_fin, &Contrat.fin.aa_fin, &Contrat.count);
        if(Vrech==Contrat.id_Voiture) {
            continue;  /* pour ne pas ecrire les info de voiture dans le new  ficher comme on supprimer*/
        }
        fprintf(N, "%f,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,\n",
            Contrat.num_contrat, Contrat.id_Voiture, Contrat.id_client,
            Contrat.debut.jj_debut, Contrat.debut.mm_debut, Contrat.debut.aa_debut,
            Contrat.fin.jj_fin, Contrat.fin.mm_fin, Contrat.fin.aa_fin, Contrat.count);
    }while(!feof(F));

    fclose(F);
    fclose(N);
    remove("ContratsLocations.txt");
    rename("nouveu_Contrat.txt", "ContratsLocations.txt");
```

❹Modifier le contrat :

C’est le client veut modifier les informations de contrat.

En demande ou client saisie numéro de contrat à modifier,

```
void Modifier_contrat(){
     gtk_widget_destroy(window5);
      system("cls");
    float num_contrat;
    contrat Contrat;
    int test = 0;


    printf("Entrez le num de contrat pour la modifier: ");
    scanf("%d",&num_contrat);

    FILE* F;
    FILE* N;
    F=fopen("ContratsLocations.txt", "r");
    N=fopen("nouveu_Contrat.txt", "w"); /* new fichier */
    do {
        fscanf(F, "%f,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,\n",
            &Contrat.num_contrat, &Contrat.id_Voiture, &Contrat.id_client,
            &Contrat.debut.jj_debut, &Contrat.debut.mm_debut, &Contrat.debut.aa_debut,
            &Contrat.fin.jj_fin, &Contrat.fin.mm_fin, &Contrat.fin.aa_fin, &Contrat.count);
        /* if le contrat exist ---> modifier la dat de fin et le cout*/
        if(num_contrat==Contrat.num_contrat) {
            test = 1;  /*pour connaitre if le contrat exist*/

            printf("\t\t\Entrez la nouvelle date de fin : ");
            scanf("%d%d%d", &Contrat.fin.jj_fin, &Contrat.fin.mm_fin,&Contrat.fin.aa_fin);
            fflush(stdin);
            printf("\t\t\tEntrez le nouveu cout: ");
             scanf("%d",&Contrat.count);
        }
        /* copier dans "ContratsLocations.txt" avec modification   */
        fprintf(N, "%f,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,\n",
            Contrat.num_contrat, Contrat.id_Voiture, Contrat.id_client,
            Contrat.debut.jj_debut, Contrat.debut.mm_debut, Contrat.debut.aa_debut,
            Contrat.fin.jj_fin, Contrat.fin.mm_fin, Contrat.fin.aa_fin, Contrat.count);
    }while(!feof(F));

    fclose(F);
    fclose(N);
    remove("ContratsLocations.txt");
    rename("nouveu_Contrat.txt", "ContratsLocations.txt");

    if(test) {
        printf("\n la modification de contrat a reussi");
    }
    else {
        printf("Le contrat n'exist pas");
    }
        Gestion_location();
}
```

❺Supprimer contrat ⏏ :

En demande ou client saisie numéro de contrat à supprimer

, et on ouvrir le fichier « contrat.txt » pour lire toutes les lignes de fichier, s'il termine va « gestion location ».

```
void supprimer_contrat(){
    gtk_widget_destroy(window5);
    system("cls");
    float nc;
    char cf_cn;
   printf("entrez le num de contrat pour le suprimer :");
   scanf("%d",&nc);
   if(recherche_contrat(nc) == 1)
   {
       printf("confermer par 'o' c'est oui ou pour annuller par 'n' c'est non : ");
       scanf("%c",&cf_cn);
       if (cf_cn=='o'||cf_cn=='O')
       {
           FILE *F, *f;
           F=fopen("contrat.txt","r");
           f=fopen("contrat1.txt","a");
           do
           {
                   fscanf(F,"f, %d, %d, %d, %d, %d \n" ,&cl.num_contrat,&cl.id_Voiture,&cl.id_client,&cl.debut,&cl.fin,&cl.count);
            if(cf_cn != cl.num_contrat)
            {
                   fprintf(f,"%f, %d, %d, %d, %d, %d \n" , cl.num_contrat,cl.id_Voiture,cl.id_client,cl.debut,cl.fin,cl.count);
            }
           }while(feof(F));
           fclose(f);
           fclose(F);
           remove("contrat.txt");
           rename("contrat1.txt","contrat.txt");
           printf("suppresion de contrat effectuee avec succes");
       }
   }
   else printf("le num de contrat n' exist pas");
   Gestion_location();
}
```

→Gestion de voiture 🚘 :

❶Liste voiture 📝 :

Permet d'afficher liste de voiture.

```
void list_voiture()
{
      gtk_widget_destroy(window3);
      system("cls");
      FILE *F;
      F=fopen("voiture.txt","r");
      do
      {
            fscanf(F,"%d,%s,%s,%s,%d,%d,%s \n" ,&V.id_voiture,&V.marque,&V.nom_voiture,&V.couleur,&V.nbplace,&V.prix_jour,&V.enlocation);
            fflush(stdin);
            printf("%d\t", V.id_voiture);
            printf("%s\t",V.marque);
            printf("%s\t",V.nom_voiture);
            printf("%s\t",V.couleur);
            printf("%d\t", V.nbplace);
            printf("%d\t", V.prix_jour);
            printf("%s\n",V.enlocation);

      }while(feof(F));
      fclose(F);

      Gestion_voiture();

}
```

Ouvrir le fichier avec le mode de lecture et après on affiche tous information de structure ligne par ligne jusqu'à la fin, s'il termine va « gestion voiture».

❷Ajouter voiture 📝 :

Permet d'ajouter les attributs de liste voiture.

```
void ajoute_voiture()
{
    gtk_widget_destroy(window3);
    system("cls");
  FILE *F;
  int id;
  F=fopen("voiture.txt","a");
  printf("\t\t\tid de nouveau voiture :");
  scanf("%d",&id);
  fflush(stdin);
  while (recherche_voiture(id)==1)
  {
      printf("l id de voiture est dejat existe ");
      printf("id de nouveau voiture :");
      scanf("%d",&id);
  }
  V.id_voiture=id;
  printf("\t\t\tentrez la marque de voiture ");
  fgets(V.marque , sizeof(V.marque),stdin);
    fflush(stdin);
  printf("\t\t\tentrez la nom de voiture ");
  fgets(V.nom_voiture , sizeof(V.nom_voiture ),stdin);;
  fflush(stdin);
  printf("\t\t\tentrez la couleur de voiture ");
  fgets(V.couleur , sizeof(V.couleur),stdin);
  fflush(stdin);
  printf("\t\t\tentrez la nbplace de voiture ");
  scanf("%d",&V.nbplace);
  fflush(stdin);
  printf("\t\t\tentrez la prixjour de voiture ");
  scanf("%d",&V.prix_jour);
  fflush(stdin);
  printf("\t\t\tentrez l enlocation de voiture ");
    fgets(V.enlocation , sizeof(V.enlocation),stdin);
    fflush(stdin);
  fprintf(F,"%d, %s, %s, %s, %d, %d, %s \n" , V.id_voiture,V.marque,V.nom_voiture,V.couleur,V.nbplace,V.prix_jour,V.enlocation);
  fclose(F);
  Gestion_voiture();
}
```

Demande de saisie id voiture à ajouter s'il existe déjà va afficher de saisie nouveau id voiture jusqu'à la fonction recherche voiture retourner -1 et après saisie les informations qui demande dans la consule, s'il termine va « gestion voiture ».

❸Modifier voiture 🗒 :

Permet de modifier les attributs de liste voiture.

```
void modifier_voiture()
{
     system("cls");
     gtk_widget_destroy(window3);
    FILE *F,*f;
    int id_mdf;
    char cf_mdf;


    printf("entrez l'id_voiture pour modifier : \n");
    scanf("%d",&id_mdf);
    if(recherche_voiture(id_mdf) == 1)
    {
        printf("\n confermer par 'o' c'est oui ou pour annuller par 'n' c'est non : ");
        scanf("%c",&cf_mdf);
        if (cf_mdf=='o'||cf_mdf=='O')
        {
            F=fopen("voiture.txt","r");
            f=fopen("voiture1.txt","a");
            do
            {
                fscanf(F,"%d, %s, %s, %s, %d, %d, %s \n" ,&V.id_voiture,&V.marque,&V.nom_voiture,&V.couleur,&V.nbplace,&V.prix_jour,&V.enlocation);
                if(id_mdf==V.id_voiture)
                {
                        V.id_voiture=id_mdf;
                            printf("\t\t\tentrez la marque de voiture ");
                            fgets(V.marque ,sizeof(V.marque),stdin);
                            printf("\n\t\t\tentrez la nom de voiture ");
                            fgets(V.nom_voiture , sizeof(V.nom_voiture ),stdin);;
                            fflush(stdin);
                            printf("\n\t\t\tentrez la couleur de voiture ");
                            fgets(V.couleur , sizeof(V.couleur),stdin);
                            fflush(stdin);
                            printf("\n\t\t\tentrez la nbplace de voiture ");
                            scanf("%d",&V.nbplace);
                            fflush(stdin);
                            printf("\n\t\t\tentrez la prixjour de voiture ");
                            scanf("%d",&V.prix_jour);
                            fflush(stdin);
                            printf("\n\t\t\tentrez l enlocation de voiture ");
                            fgets(V.enlocation , sizeof(V.enlocation),stdin);
                            fflush(stdin);
                }
                fprintf(f,"%d,%s,%s,%s,%d,%d,%s/n" ,V.id_voiture,V.marque,V.nom_voiture,V.couleur,V.nbplace,V.prix_jour,V.enlocation);
            }while(feof(F));
            fclose(f);
            fclose(F);
            remove("voiture.txt");
            rename("voiture1.txt","voiture.txt");
            printf("\n la modification de voiture a reussi");
        }
        else printf("modification annuler ");
    }
    else printf("id de voiture n'exist pas");
    Gestion_voiture();
}
```

Demande de saisie id voiture à modifier s'il existe déjà, modifier les informations qui demande dans la consule, s'il termine va à « gestion voiture », sinon va afficher qu'id n'existe pas.

❹Supprimer voiture ✂ :

Permet de supprimer les attributs de liste voiture.

```
void suprimer_voiture()
{
     system("cls");
     gtk_widget_destroy(window3);
    int id_sup;
    char cf_sup;
    printf("entrez le id de voiture pour le suprimer :");
    scanf("%d",&id_sup);
    if(recherche_voiture(id_sup) == 1)
    {
        printf("confermer par 'o' c'est oui ou pour annuller par 'n' c'est non : ");
        scanf("%c",&cf_sup);
        if (cf_sup=='o'||cf_sup=='O')
        {
            FILE *F, *f;
            F=fopen("voiture.txt","r");
            f=fopen("voiture1.txt","a");
            do
            {
                fscanf(F,"%d,%s,%s,%s,%d,%d,%s/n" ,&V.id_voiture,&V.marque,&V.nom_voiture,&V.couleur,&V.nbplace,&V.prix_jour,&V.enlocation);
                if(id_sup != V.id_voiture)
                {
                    fprintf(f,"%d,%s,%s,%s,%d,%d,%s\n" ,V.id_voiture,V.marque,V.nom_voiture,V.couleur,V.nbplace,V.prix_jour,V.enlocation);
                }
            }while(feof(F));
            fclose(f);
            fclose(F);
            remove("voiture.txt");
            rename("voiture1.txt","voiture.txt");
            printf("suppresion effectuee avec succes");
        }
    }
```

Demande de saisie id voiture à supprimer s'il existe déjà pour ça on utilise un notre fichier, sinon va afficher qu'id n'existe pas, s'il termine va à « gestion voiture ».

→Gestion de client 👤 :

❶Liste client 📝 :

Permet d'afficher une liste de client.

```
void list_Clients()
{
    system("cls");
    gtk_widget_destroy(window4);
    FILE *F;
    F=fopen("Client.txt","r");

    do
    {
        fscanf(F,"%d, %s, %s, %d, %s, %d \n" ,&Client.id_client,&Client.nom,&Client.prenom,&Client.cin,&Client.adresse,&Client.telephone);
        fflush(stdin);
        printf("%d\t",Client.id_client);
        printf("%s\t",Client.nom);
        printf("%s\t",Client.prenom);
        printf("%d\t",Client.cin);
        printf("%s\t",Client.adresse);
        printf("%d\t",Client.telephone);
    }while(feof(F));
    fclose(F);
    Gestion_client();

}
```

Ouvrir le fichier avec le mode de lecture et après on affiche tous information de structure ligne par ligne jusqu'à la fin, s'il termine va « gestion client ».

❷Ajouter client 📝 :

Permet d'ajouter les attributs de liste client.

```
void ajoute_Client()
{
    system("cls");
    gtk_widget_destroy(window4);
    FILE *F;
    int id;
    F=fopen("Client.txt","a");
    printf("id de nouveau Client :");
    scanf("%d",&id);
    fflush(stdin);
    while (recherche_Client(id)==1)
    {
        printf("l'id de Client est dejat existe ");
        printf("l'id de nouveau Client :");
        scanf("%d",&id);
    }
    Client.id_client=id;
    printf("\t\t\tentrez le Nom de Client : ");
    fgets(Client.nom , sizeof(Client.nom),stdin);
    fflush(stdin);
    printf("\t\t\tentrez le Prenom de Client : ");
    fgets(Client.prenom,sizeof(Client.prenom),stdin);
    fflush(stdin);
    printf("\t\t\tentrez le CIN de Client : ");
    scanf("%d",&Client.cin);
    fflush(stdin);
    printf("\t\t\tentrez l'adresse de Client : ");
    fgets(Client.adresse,sizeof(Client.adresse),stdin);
    fflush(stdin);
    printf("\t\t\tentrez N° de telephone de Client : ");
    scanf("%d",&Client.telephone);
    fflush(stdin);
    fprintf(F,"%d, %s, %s, %d, %s, %d /n" , Client.id_client,Client.nom,Client.prenom,Client.cin,Client.adresse,Client.telephone);
    fclose(F);
    Gestion_client();
}
```

Demande de saisie id client à ajouter s'il existe déjà va afficher de saisie nouveau id client jusqu'à la fonction recherche client retourner -1 et après saisie les informations qui demande dans la consule, s'il termine va « gestion client ».

❸Modifier client 🗒 :

Permet de modifier les attributs de liste client.

```c
void modifier_Client()
{
    system("cls");
    gtk_widget_destroy(window4);
   int id_mdf;
   char cf_mdf;

   printf("entrez l'id de client pour modifier :");
   scanf("%d",&id_mdf);
   if(recherche_Client(id_mdf) == 1)
   {
       printf("\n confermer par  'o' c'est oui ou pour annuller par  'n' c'est non : ");
       scanf("%c",&cf_mdf);
       if (cf_mdf=='o'||cf_mdf=='O')
       {
           FILE *F,*f;
           F=fopen("Client.txt","r");
           f=fopen("client1.txt","a");
           do
           {
                   fscanf(F,"%d, %s, %s, %d, %s, %d \n" ,&Client.id_client,&Client.nom,&Client.prenom,&Client.cin,&Client.adresse,&Client.telephone);
                   if(id_mdf==Client.id_client)
                   {
                     Client.id_client=id_mdf;
                                 printf("\t\t\tentrez le Nom de Client : ");
                                   fgets(Client.nom , sizeof(Client.nom),stdin);
                                   fflush(stdin);
                                   printf("\t\t\tentrez le Prenom de Client : ");
                                   fgets(Client.prenom,sizeof(Client.prenom),stdin);
                                   fflush(stdin);
                                   printf("\t\t\tentrez le CIN de Client : ");
                                   scanf("%d",&Client.cin);
                                   fflush(stdin);
                                   printf("\t\t\tentrez l'adresse de Client : ");
                                   fgets(Client.adresse,sizeof(Client.adresse),stdin);
                                   fflush(stdin);
                                   printf("\t\t\tentrez N° de telephone de Client : ");
                                   scanf("%d",&Client.telephone);
                                   fflush(stdin);
               }
               fprintf(f,"%d, %s, %s, %d, %s, %d \n" , Client.id_client,Client.nom,Client.prenom,Client.cin,Client.adresse,Client.telephone);
           }while(feof(F));
           fclose(f);
           fclose(F);
           remove("Client.txt");
           rename("Client1.txt","Client.txt");
           printf("\n la modification de client a reussi");
       }
       else printf("modification annuler ");
   }
   else printf("l'id n'exist pas");
   Gestion_client();
}
```

Demande de saisie id client à modifier s'il existe déjà, modifier les informations qui demande dans la consule, s'il termine va à « gestion client », sinon va afficher qu'id n'existe pas.

❹Supprimer client ✂ :

Permet de supprimer les attributs de liste client.

```
void suprimer_client()
{
    system("cls");
    gtk_widget_destroy(window4);
    system("cls");
   int id_sup;
   char cf_sup;
   printf("entrez le id de Client pour le suprimer :");
   scanf("%d",&id_sup);
   if(recherche_Client(id_sup) == 1)
   {
       printf("confermer par 'o' c'est oui ou pour annuller par 'n' c'est non : ");
       scanf("%c",&cf_sup);
       if (cf_sup=='o'||cf_sup=='O')
       {
           FILE *F, *f;
           F=fopen("Client.txt","r");
           f=fopen("Client1.txt","a");
           do
           {
                   fscanf(F,"%d, %s, %s, %d, %s, %d \n" ,&Client.id_client,&Client.nom,&Client.prenom,&Client.cin,&Client.adresse,&Client.telephone);
               if(id_sup != Client.id_client)
               {
                       fprintf(f,"%d, %s, %s, %d, %s, %d \n" , Client.id_client,Client.nom,Client.prenom,Client.cin,Client.adresse,Client.telephone);
               }
           }while(!feof(F));
           fclose(f);
           fclose(F);
           remove("Client.txt");
           rename("Client1.txt","voiture.txt");
           printf("suppresion de client effectuee avec succes");
       }
   }
   else printf("l' id_voiture n' exist pas");

   Gestion_client();
}
```

Demande de saisie id client à supprimer s'il existe déjà pour ça on utilise un notre fichier, sinon va afficher qu'id n'existe pas, s'il termine va à « gestion client ».