

# Rapport du projet programmation

1ere annee Genie des systemes Infotronique  
Groupe A

## Gestion d'une station de recharge des voitures électriques

Réalisé par :  
Mhamdi Becem  
Jendoubi Nour

# Tables des Matieres

Introduction.....	2
Les Structures.....	3
Les Prototypes.....	4
Menu et sous menus.....	5
Les fonctions d'allocations.....	14
Saisie du station.....	15
Saisie du Client.....	16
Saisie du Chargeur.....	17
Affichage du Client.....	18
Affichage du Chargeur.....	19
Affichage du Voiture.....	20
Affichage du Paiement.....	21
<b>Traitement:</b>	
Traitement : Total du paiement.....	22
Traitement : Temps restant pour charger une voiture.....	22
Traitement : Trouver un chargeur disponible et de même type.....	23
Traitement : Combien de chargeurs disponible dans la station.....	23
Traitement : Moin chargée voiture.....	24
Traitement : Plus chargée voiture.....	25
Traitement : Marque de voiture la plus fréquente.....	26
<b>Exécution:</b>	
Exécution modifier client.....	27
Exécution ajout client.....	28
Exécution supprimer client.....	29
Exécution modifier chargeur.....	30
Exécution ajout chargeur.....	31
Exécution supprimer chargeur.....	32
Exécution modifier voiture.....	33
Exécution ajout voiture.....	34
Exécution supprimer voiture.....	35
Exécution modifier paiement.....	36
Exécution ajout paiement.....	37
Exécution supprimer paiement.....	38
<b>Resultat:</b>	
Execution tableau dynamique des résultats.....	39
QR code .....	40

# **Introduction:**

**Bienvenue dans notre projet de gestion d'une station de recharge des voitures électriques. Cette plateforme offre une solution efficace pour la gestion des Clients et des Chargeurs, avec des fonctionnalités clés telles que l'ajout rapide de nouveaux Client , Chargeur, Voiture et paiement, la prise de rendez-vous simplifiée, la suppression facile des clients, chargeurs,voitures,paiements et la modification rapide des informations des clients,des chargeurs,des paiements ou des voitures.**

**avec notre outil intuitif et optimisez la gestion de vos recharges et de vos paiements.**

# 2 Les Structures:

```
struct resultat{
    int CodeClient;
    float SommePaielements;
};typedef struct resultat RESULTAT;
```

You, 16 hours ago | 1 author (You)

```
struct voiture {
    int NbSerie;
    char Marque[100];
    char Model[100];
    float Pourcentage;
    int TypeChargeur;
};
typedef struct voiture VOITURE;
```

You, 2 days ago | 1 author (You)

```
struct date {
    int jour;
    int mois;
    int annee;
};
typedef struct date DATE;
```

You, 16 hours ago | 1 author (You)

```
struct paiement {
    DATE date;
    int prix;
};
typedef struct paiement PAIEMENT;
```

You, 16 hours ago | 1 author (You)

```
struct client {
    int NbVoiture;
    VOITURE* voiture;
    int NbPaiement;
    PAIEMENT* paiement;
};
typedef struct client CLIENT;
```

```
struct chargeur {
    int EtatUtilisation;
    int TypeChargeur;
};
typedef struct chargeur CHARGEUR;
```

You, 2 days ago | 1 author (You)

```
struct station {
    int NbChargeur;
    int NbClient;
    CLIENT* client;
    CHARGEUR* chargeur;
};
typedef struct station STATION;
```

# Les Prototypes:

```
void RemplireStation(STATION*);
void AllocationChargeurs(CHARGEUR**,int);
void AllocationClients(CLIENT**,int);
void AllocationVoitures(VOITURE**,int);
void AllocationPaielements(PAIEMENT**,int);
void RemplireClient(STATION**,int );
void RemplireChargeur(STATION**,int );
void RemplireFichierduTableau(STATION*);
void RemplireTableauDuFichier(STATION*);
void LibereMemoire(STATION*);
void ModifierClient(STATION*);
void ModifierChargeur(STATION*);
void ModifierVoiture(STATION*);
void ModifierPaielement(STATION*);
void AjoutClient(STATION*);
void AjoutChargeur(STATION*);
void AjoutVoiture(STATION*);
void AjoutPaielement(STATION*);
void SupprimerChargeur(STATION*);
void SupprimerClient(STATION*);
void SupprimerVoiture(STATION*);
void SupprimerPaielement(STATION*);
void AfficherClient(STATION*);
void AfficherChargeur(STATION*);
void AfficherPaielement(STATION*);
void AfficherVoiture(STATION*);
float TotalePaielement(STATION*);
void TrouverChargeur(STATION*);
void CombienChargeurDisponible(STATION*);
void TempsRestant(STATION*);
void PlusChargeeVoiture(STATION*);
void MoinChargeeVoiture(STATION*);
void MarqueDeVoiture(STATION*);
RESULTAT* MeilleurClient(STATION*);
```

# Menu et sous menus:

```
int ChoixMenu;
```

```
do { You, 16 hours ago • Traitement...
```

```
{
    printf("\n*----- Menu Principal -----*\n");
    printf("1. Modification du fichier\n");
    printf("2. Traitement\n");
    printf("3. Affichage\n");
    printf("9. Quitter\n");
    printf("Votre choix : ");
    scanf("%d", &ChoixMenu);
    switch (ChoixMenu)
    {
    case 1:
        int ChoixModification;
        do
        {
            printf("\n*----- Modification du Fichier -----*\n");
            printf("1. Modifier Client\n");
            printf("2. Modifier Chargeur\n");
            printf("3. Modifier Paiement\n");
            printf("4. Modifier Voiture\n");
            printf("9. Retour au menu principal\n");
            printf("Votre choix : ");
            scanf("%d", &ChoixModification);
            switch (ChoixModification)
            {
            case 1:
                int ChoixClient;
                do
                {
                    printf("\n*-----Modification du Client-----*\n");
                    printf("1. Modifier un Client\n");
                    printf("2. Ajout un Client\n");
                    printf("3. Supprimer un Client\n");
                    printf("9. Retour au menu principal\n");
                    printf("Votre choix : ");
                    scanf("%d", &ChoixClient);
                    switch (ChoixClient)
                    {
                    case 1:
                        printf("1. Modifier un Client\n");
                        printf("2. Ajout un Client\n");
                        printf("3. Supprimer un Client\n");
                        printf("9. Retour au menu principal\n");
                        printf("Votre choix : ");
                        scanf("%d", &ChoixClient);
                    case 2:
                        printf("Ajout un Client\n");
                        printf("9. Retour au menu principal\n");
                        printf("Votre choix : ");
                        scanf("%d", &ChoixClient);
                    case 3:
                        printf("Supprimer un Client\n");
                        printf("9. Retour au menu principal\n");
                        printf("Votre choix : ");
                        scanf("%d", &ChoixClient);
                    }
                } while (ChoixClient != 9);
            case 2:
                printf("Ajout un Client\n");
                printf("9. Retour au menu principal\n");
                printf("Votre choix : ");
                scanf("%d", &ChoixClient);
            case 3:
                printf("Supprimer un Client\n");
                printf("9. Retour au menu principal\n");
                printf("Votre choix : ");
                scanf("%d", &ChoixClient);
            case 4:
                printf("Ajout un Client\n");
                printf("9. Retour au menu principal\n");
                printf("Votre choix : ");
                scanf("%d", &ChoixClient);
            }
        } while (ChoixModification != 9);
    }
}
```

```

        printf("9. Retour au menu modification\n");
        printf("Votre choix : ");
        scanf("%d",&ChoixClient);
        switch (ChoixClient)
        {
        case 1:
            printf("\n*-----Modifier un client-----*\n");
            ModifierClient(station);
            break;
        case 2:
            printf("\n*-----Ajout un client-----*\n");
            AjoutClient(station);
            break;
        case 3:
            printf("\n*-----supprimer un client-----*\n");
            SupprimerClient(station);
            break;
        default: (char [25])"Tapez un Nombre Valide!!"
            printf("Tapez un Nombre Valide!!");
            break;
        }
    } while (ChoixClient !=9);
    break;
case 2:
    int ChoixChargeur;
    do
    {
        printf("\n*-----Modification du Chargeur-----*\n");
        printf("1. Modifier un Chargeur\n");
        printf("2. Ajout un Chargeur\n");
        printf("3. Supprimer un Chargeur\n");
        printf("9. Retour au menu modification\n");
        printf("Votre choix : ");
        scanf("%d",&ChoixChargeur);
        switch (ChoixChargeur)
        {
            case 1:

```

```

        case 1:
            printf("\n*-----Modification du Chargeur-----*\n");
            ModifierChargeur(station);
            break;
        case 2:
            printf("\n*-----Ajout un Chargeur-----*\n");
            AjoutChargeur(station);
            break;
        case 3 :
            printf("\n*-----Suppression du Chargeur-----*\n");
            SupprimerChargeur(station);
            break;
        default:
            printf("Tapez un Nombre Valide!!");
            break;
    }
} while (ChoixChargeur !=9);
break;
case 3:
    int ChoixPaielement;
    do
    {
        printf("\n*-----Modification du Paiement-----*\n");
        printf("1. Modifier un Paiement\n");
        printf("2. Ajout un Paiement\n");
        printf("3. Supprimer un Paiement\n");
        printf("9. Retour au menu Paiement\n");
        printf("Votre choix : ");
        scanf("%d",&ChoixPaielement);
        switch (ChoixPaielement)
        {
            case 1:
                printf("\n*-----Modification du Paiement-----*\n");
                ModifierPaiement(station);
                break;
            case 2:
                printf("\n*-----Ajout du Paiement-----*\n");
                AjoutPaiement(station);
                break;
            case 3:
                printf("\n*-----Suppression du Paiement-----*\n");
                SupprimerPaiement(station);
                break;
            default:
                printf("Tapez un Nombre Valide!!");
                break;
        }
    } while (ChoixPaielement !=9);
    break;
}

```



```

        case 2:
            printf("\n*-----Ajout du Paiement-----*\n");
            AjoutPaiement(station);
            break;
        case 3:
            printf("\n*-----Suppression du Paiement-----*\n");
            SupprimerPaiement(station);
            break;
        default:
            printf("Tapez un Nombre Valide!!");
            break;
    }
} while (ChoixPaiement != 9);
break;

case 4:
    int ChoixVoiture;
    do
    {
        printf("\n*-----Modification du Voiture-----*\n");
        printf("1. Modifier un Voiture\n");
        printf("2. Ajout un Voiture\n");
        printf("3. Supprimer un Voiture\n");
        printf("9. Retour au menu Voiture\n");
        printf("Votre choix : ");
        scanf("%d",&ChoixVoiture);
        switch (ChoixVoiture)
        {
            case 1:
                printf("\n*-----Modification du Voiture-----*\n");
                ModifierVoiture(station);
                break;
            case 2:
                printf("\n*-----Ajout du Voiture-----*\n");
                AjoutVoiture(station);
                break;
            case 3 :
                printf("\n*-----Suppression du Voiture-----*\n");

```

```

        case 3 :
            printf("\n*-----Suppression du Voiture-----*\n");
            SupprimerVoiture(station);
            break;

        default:
            break;
    }
} while (ChoixVoiture != 9);
break;
default:
    printf("Tapez un Nombre Valide!!");
    break;
}
} while (ChoixModification != 9);
break;
case 2:
    int ChoixTraitement;
    do
    {
        printf("\n*-----Traitement-----*\n");
        printf("1. Totale des paiement\n");
        printf("2. Temps restant pour charger une voiture\n");
        printf("3. Trouver un chargeur disponible et de type conforme\n");
        printf("4. Combien de chargeur disponible\n");
        printf("5. Moin Chargee voiture\n");
        printf("6. Plus chargee voiture\n");
        printf("7. Marque de la voiture la plus frequente\n");
        printf("8. Meilleur Client(Resultat)\n");
        printf("9. Retour au menu\n");
        printf("Votre choix : ");
        scanf("%d",&ChoixTraitement);
        switch (ChoixTraitement)
        {
            case 1:
                float x;
                x = TotalePaielement(station);

```

```

case 1:
    float x;
    x = TotalePaielement(station);
    printf("Le totale des paiement est : %.2f",x);
    break;
case 2:
    TempsRestant(station);
    break;
case 3:
    TrouverChargeur(station);
    break;
case 4:
    CombienChargeurDisponible(station);
    break;
case 5:
    PlusChargeeVoiture(station);
    break;
case 6:
    MoinChargeeVoiture(station);
    break;
case 7:
    MarqueDeVoiture(station);
    break;
case 8:
    RESULTAT *resultat;
    resultat = MeilleurClient(station);
    printf("les 3 meilleurs clients sont : \n");
    for(int i =0;i<3;i++){
        printf("%d\n",resultat->CodeClient);
    }
    break;
default:
    printf("Tapez un Nombre Valide!!");
    break;
}

} while (ChoixTraitement != 9);

```

```

        } while (ChoixTraitement != 9);

        break;
case 3:
int ChoixAffichage;
do
{
    printf("\n*-----Affichage-----*\n");
    printf("1. Affichage Client\n");
    printf("2. Affichage Chargeur\n");
    printf("3. Affichage Voiture\n");
    printf("4. Affichage Paiement\n");
    printf("9. Retour au menu\n");
    printf("Votre choix : ");
    scanf("%d",&ChoixAffichage);
    switch (ChoixAffichage)
    {
        case 1:
            AfficherClient(station);
            break;
        case 2:
            AfficherChargeur(station);
            break;
        case 3 :
            AfficherVoiture(station);
            break;
        case 4:
            AfficherPaiement(station);
            break;

        default:
            printf("Tapez un Nombre Valide!!");
            break;
    }
} while (ChoixAffichage !=9);

```

```
    default:  
        printf("GoodBye");  
        break;  
    }  
} while (ChoixMenu != 9);
```

# Execution:

## Menu Principal:

```
*----- Menu Principal -----*
1. Modification du fichier
2. Traitement
3. Affichage
9. Quitter
Votre choix :
```

## 1.Modification du fichier:

```
*----- Modification du Fichier -----*
1. Modifier Client
2. Modifier Chargeur
3. Modifier Paiement
4. Modifier Voiture
9. Retour au menu principal
Votre choix :
```

### 1.1.Modifier Client:

```
*-----Modification du Client-----
1. Modifier un Client
2. Ajout un Client
3. Supprimer un Client
9. Retour au menu modification
Votre choix :
```

### 1.2.Modifier Chargeur

```
*-----Modification du Chargeur-----*
1. Modifier un Chargeur
2. Ajout un Chargeur
3. Supprimer un Chargeur
9. Retour au menu modification
Votre choix :
```

## 1.3.Modifier Paiement:

```
*-----Modification du Paiement-----*
1. Modifier un Paiement
2. Ajout un Paiement
3. Supprimer un Paiement
9. Retour au menu Paiement
Votre choix :
```

## 1.4.Modifiée Voiture:

```
*-----Modification du Voiture-----*
1. Modifier un Voiture
2. Ajout un Voiture
3. Supprimer un Voiture
9. Retour au menu Voiture
Votre choix :
```

## 2. Traitement:

```
*-----Traitement-----*
1. Totale des paiement
2. Temps restant pour charger une voiture
3. Trouver un chargeur disponible et de type conforme
4. Combien de chargeur disponible
5. Moin Chargee voiture
6. Plus chargee voiture
7. Marque de la voiture la plus frequente
8. Meilleur Client(Resultat)
9. Retour au menu
Votre choix :
```

## 3. Affichage:

```
*-----Affichage-----*
1. Affichage Client
2. Affichage Chargeur
3. Affichage Voiture
4. Affichage Paiement
9. Retour au menu
Votre choix :
```

# Les Fonction d'allocations:

```
void AllocationChargeurs(CHARGEUR** chargeur, int n) {
    *chargeur = (CHARGEUR*)malloc(n * sizeof(CHARGEUR));
    if (!*chargeur) {
        printf("Erreur d'allocation memoire pour les chargeurs.");
        exit(-1);
    }
}

void AllocationClients(CLIENT** client, int n) {
    *client = (CLIENT*)malloc(n * sizeof(CLIENT));
    if (!*client) {
        printf("Erreur d'allocation memoire pour les clients.");
        exit(-2);
    }
}

void AllocationVoitures(VOITURE**voiture,int n){
    *voiture = (VOITURE*)malloc(n * sizeof(VOITURE));
    if (!*voiture) {
        printf("Erreur d'allocation memoire pour les voitures.");
        exit(-3);
    }
}

void AllocationPaielements(PAIEMENT**paiement,int n){
    *paiement = (PAIEMENT*)malloc(n * sizeof(PAIEMENT));
    if (!*paiement) {
        printf("Erreur d'allocation memoire pour les paiements.");
        exit(-4);
    }
}
```

---

You, 2 days ago • Traitement presque finie

# Saisie du Station:

```
void RemplireStation(STATION*station){
    printf("Saisire le nombre de chargeurs : ");
    scanf("%d",&station->NbChargeur);
    printf("Saisire le nombre de clients : ");
    scanf("%d",&station->NbClient);
    AllocationChargeurs(&station->chargeur,station->NbChargeur);
    AllocationClients(&station->client,station->NbClient);
    for (int i = 0;i<station->NbChargeur;i++){
        RemplireChargeur(&station,i);
    }
    for (int i = 0;i<station->NbClient;i++){
        RemplireClient(&station,i);
    }
}
```

Saisire le nombre de chargeurs : 3

Saisire le nombre de clients : 4



# Saisie du clients:

```
void RemplireClient(STATION**station,int i){
    printf("Tapez le Nombre des voiture : ");
    scanf("%d",&(*station)->client[i].NbVoiture);
    printf("Tapez le Nombre des paiements : ");
    scanf("%d",&(*station)->client[i].NbPaiement);
    AllocationVoitures(&(*station)->client[i].voiture,(*station)->client->NbVoiture);
    AllocationPaiements(&(*station)->client[i].paiement,(*station)->client->NbPaiement);
    for (int j = 0;j<(*station)->client[i].NbVoiture;j++){
        //serie
        printf("Tapez le numero du serie du voiture : ");
        scanf("%d",&(*station)->client[i].voiture[j].NbSerie);
        //model
        printf("Tapez la marque du serie du voiture : ");
        scanf("%99s",&(*station)->client[i].voiture[j].Marque);
        //marque
        printf("Tapez le model du serie du voiture : ");
        scanf("%99s",&(*station)->client[i].voiture[j].Model);
        //pourcentage
        printf("Tapez le pourcentage du serie du voiture : ");
        scanf("%f",&(*station)->client[i].voiture[j].Pourcentage);
        printf("Saisire le type du chargeur (1 = american , 2 = francais , 3 = anglais , 4 = african) :");
        scanf("%d",&(*station)->client[i].voiture[j].TypeChargeur);
    }
    for (int j = 0;j<(*station)->client[i].NbPaiement;j++){
        //date
        printf("Tapez la date du paiements : \n");
        printf("Jour : ");
        scanf("%d",&(*station)->client[i].paiement[j].date.jour);
        printf("Mois : ");
        scanf("%d",&(*station)->client[i].paiement[j].date.mois);
        printf("Annee : ");
        scanf("%d",&(*station)->client[i].paiement[j].date.annee);
        //prix
        printf("Tapez le montant : ");
        scanf("%f",&(*station)->client[i].paiement[j].prix);
    }
}
```

You. 2 days ago • Traitement presque finie

```
Tapez le Nombre des voiture : 2
Tapez le Nombre des paiements : 2
Tapez le numero du serie du voiture : 123
Tapez la marque du serie du voiture : Tesla
Tapez le model du serie du voiture : ModelX
Tapez le pourcentage du serie du voiture : 55.4
Saisire le type du chargeur (1 = american , 2 = francais , 3 = anglais , 4 = african) :1
Tapez le numero du serie du voiture : 456
Tapez la marque du serie du voiture : Tesla
Tapez le model du serie du voiture : CyberTruck
Tapez le pourcentage du serie du voiture : 30.5
Saisire le type du chargeur (1 = american , 2 = francais , 3 = anglais , 4 = african) :3
Tapez la date du paiements :
Jour : 16
Mois : 01
Annee : 2003
Tapez le montant : 50
Tapez la date du paiements :
Jour : 23
Mois : 06
Annee : 2002
Tapez le montant : 78
```

# Saisie du chargeur:

```
void RemplireChargeur(STATION**station,int i){
    printf("Saisie l'etat d'utilisation du chargeur (1 = available , 2 = not availabe) : ");
    scanf("%d",&(*station)->chargeur[i].EtatUtilisation);
    printf("Saisire le type du chargeur (1 = american , 2 = francais , 3 = anglais , 4 = african) :");
    scanf("%d",&(*station)->chargeur[i].TypeChargeur);
}
void RemnlireClient(STATION**station,int i){
```

```
Saisie l'etat d'utilisation du chargeur (1 = available , 2 = not availabe) : 1
Saisire le type du chargeur (1 = american , 2 = francais , 3 = anglais , 4 = african) :3
Saisie l'etat d'utilisation du chargeur (1 = available , 2 = not availabe) : 1
Saisire le type du chargeur (1 = american , 2 = francais , 3 = anglais , 4 = african) :2
Saisie l'etat d'utilisation du chargeur (1 = available , 2 = not availabe) : 2
Saisire le type du chargeur (1 = american , 2 = francais , 3 = anglais , 4 = african) :4
```

# Affichage Client:

Saisir le code du client a Afficher : 1

Code Client : 1

Nombre Voiture : 2

Numero de serie de la voiture : 123

Marque de la voiture :

Tesla

Model de la voiture :

ModelX

Pourcentage de la voiture : 55.40

TypeChargeur de la voiture : 1

Numero de serie de la voiture : 456

Marque de la voiture :

Tesla

Model de la voiture :

CyberTruck

Pourcentage de la voiture : 30.50

TypeChargeur de la voiture : 3

Nombre Paiement : 2

Date du paiement : 16/ 1/2003

Montant du paiement : 0.00

Date du paiement : 23/ 6/2002

Montant du paiement : 0.00

```
void AfficherClient(STATION*station){
    int ind;
    printf("Saisir le code du client a Afficher : ");
    scanf("%d", &ind);
    ind -= 1;
    if (ind < 0 || ind >= station->NbClient) {
        printf("Code client invalide!\n");
        return;
    }
    printf("Code Client : %d\n",ind+1);
    printf("Nombre Voiture : %d\n",station->client[ind].NbVoiture);
    for (int i =0;i<station->client[ind].NbVoiture;i++){
        printf("Numero de serie de la voiture : %d\n",station->client[ind].voiture[i].NbSerie);
        printf("Marque de la voiture : %99s\n",station->client[ind].voiture[i].Marque);
        printf("Model de la voiture : %99s\n",station->client[ind].voiture[i].Model);
        printf("Pourcentage de la voiture : %.2f\n",station->client[ind].voiture[i].Pourcentage);
        printf("TypeChargeur de la voiture : %d\n",station->client[ind].voiture[i].TypeChargeur);
    }

    printf("Nombre Paiement : %d\n",station->client[ind].NbPaiement);
    for(int i = 0;i<station->client[ind].NbPaiement;i++){
        printf("Date du paiement : %2d/%2d/%4d\n",station->client[ind].paiement[i].date.jour,
            station->client[ind].paiement[i].date.mois,station->client[ind].paiement[i].date.annee);
        printf("Montant du paiement : %.2f\n",station->client[ind].paiement[i].prix);
    }
}
```

# Affichage Chargeur:

```
void AfficherChargeur(STATION*station){
    int ind;
    printf("Saisir le code du chargeur a Afficher : ");
    scanf("%d", &ind);
    ind -= 1;
    if (ind < 0 || ind >= station->NbChargeur) {
        printf("Code chargeur invalide!\n");
        return;
    }
    printf("Etat utilisation : %d\n",station->chargeur[ind].EtatUtilisation);
    printf("TypeChargeur : %d\n",station->chargeur[ind].TypeChargeur);
}
```

```
Saisir le code du chargeur a Afficher : 1
Etat utilisation : 1TypeChargeur : 1
```

# Affichage Voiture:

```
void AfficherVoiture(STATION*station){
    int indC,indV;
    printf("Saisir le code du client a Afficher : ");
    scanf("%d", &indC);
    indC -= 1;
    if (indC < 0 || indC >= station->NbClient) {
        printf("Code client invalide!\n");
        return;
    }
    printf("Saisir le code du voiture a Afficher : ");
    scanf("%d", &indV);
    indV -= 1;
    if (indV < 0 || indV >= station->client[indC].NbVoiture) {
        printf("Code Voiture invalide!\n");
        return;
    }
    printf("Numero Serie de la voiture : %d\n",station->client[indC].voiture[indV].NbSerie);
    printf("Marque de la voiture : %99s\n",station->client[indC].voiture[indV].Marque);
    printf("Model de la voiture : %99s\n",station->client[indC].voiture[indV].Model);
    printf("Pourcentage de la voiture : %.2f\n",station->client[indC].voiture[indV].Pourcentage);
    printf("TypeChargeur de la voiture : %d\n",station->client[indC].voiture[indV].TypeChargeur);
}
```

Saisir le code du client a Afficher : 1

Saisir le code du voiture a Afficher : 1

Numero Serie de la voiture : 123

Marque de la voiture :

Tesla

Model de la voiture :

ModelX

Pourcentage de la voiture : 55.40

TypeChargeur de la voiture : 1

# Affichage Paiement:

```
void AfficherPaiement(STATION*station){
    int indC,indP;
    printf("Saisir le code du client a Afficher : ");
    scanf("%d", &indC);
    indC -= 1;
    if (indC < 0 || indC >= station->NbClient) {
        printf("Code client invalide!\n");
        return;
    }
    printf("Saisir le code du paiement a Afficher : ");
    scanf("%d", &indP);
    indP -= 1;
    if (indP < 0 || indP >= station->client[indC].NbPaiement) {
        printf("Code Paiement invalide!\n");
        return;
    }
    printf("Date du paiement : %2d/%2d/%4d\n",station->client[indC].paiement[indP].date.jour,
station->client[indC].paiement[indP].date.mois,station->client[indC].paiement[indP].date.annee);
    printf("Montant du paiement : %.2f \n",station->client[indC].paiement[indP].prix);
}
```

```
Saisir le code du client a Afficher : 1
Saisir le code du paiement a Afficher : 2
Date du paiement : 23/ 6/2002
Montant du paiement : 0.00
```

# Traitements:

## 1. Totale des paiements:

```
float TotalePaielement(STATION*station){
    float x = 0;
    for (int i = 0;i<station->NbClient;i++){
        for (int j = 0;j<station->client[i].NbPaielement;j++){
            x += station->client[i].paielement[j].prix;
        }
    }
    return x;
}
```

Le totale des paielement est : 2225340416.00

## 2. Temps restant pour charger une voiture:

```
void TempsRestant(STATION*station){
    int indC,indV;
    printf("Saisir le code du client : ");
    scanf("%d", &indC);
    indC -= 1;
    if (indC < 0 || indC >= station->NbClient) {
        printf("Code client invalide!\n");
        return;
    }
    printf("Saisir le code du voiture : ");
    scanf("%d", &indV);
    indV -= 1;
    if (indV < 0 || indV >= station->client[indC].NbVoiture) {
        printf("Code Voiture invalide!\n");
        return;
    }
    float temp;
    int heure,minute,seconde;
    temp = (100 - station->client[indC].voiture[indV].Pourcentage)*150;
    heure = (int)temp / 3600;
    minute = ((int)temp %3600 )/60;
    seconde = (int)temp %60;
    printf("Temps restant est : %d:%d:%d",heure,minute,seconde);
}
```

```
Saisir le code du client : 1
Saisir le code du voiture : 2
Temps restant est : 2:53:45
```

### 3.Trouver un chargeur disponible:

```
void TrouverChargeur(STATION*station){
    int indC,indV;
    printf("Saisir le code du client : ");
    scanf("%d", &indC);
    indC -= 1;
    if (indC < 0 || indC >= station->NbClient) {
        printf("Code client invalide!\n");
        return;
    }
    printf("Saisir le code du voiture : ");
    scanf("%d", &indV);
    indV -= 1;
    if (indV < 0 || indV >= station->client[indC].NbVoiture) {
        printf("Code Voiture invalide!\n");
        return;
    }
    int NbChargeurDisponible= 0;
    int TypeChargeurVoiture = station->client[indC].voiture[indV].TypeChargeur;
    for (int i = 0;i<station->NbChargeur;i++){
        if (TypeChargeurVoiture == station->chargeur[i].TypeChargeur && station->chargeur[i].EtatUtilisation == 1)
            NbChargeurDisponible ++;
    }
    if (NbChargeurDisponible>0) printf("Il est %d Chargeur disponible de meme type \n",NbChargeurDisponible);
    else printf("Il n'ya pas de chargeur de meme type disponible \n");
}
```

```
Saisir le code du client : 1
Saisir le code du voiture : 2
Il n'ya pas de chargeur de meme type disponible
```

```
Saisir le code du client : 1
Saisir le code du voiture : 1
Il est 1 Chargeur disponible de meme type
```



## 4. Combien de chargeur disponible dans la station:

```
void CombienChargeurDisponible(STATION*station){
    int disp;
    for (int i =0;i<station->NbChargeur;i++){
        if(station->chargeur[i].EtatUtilisation == 1) disp++;
    }
    if (disp>0) printf("Il y'a %d Chargeur disponible\n",disp);
    else printf("Il n'ya aucun chargeur disponible\n");
}
```

Votre choix : 4

Il y'a 2 Chargeur disponible

## 5. Voiture moins chargée:

```
void MoinChargeeVoiture(STATION*station){
    float min = 0;
    int indMinC= 0,indMinV = 0;

    for (int i = 0;i<station->NbClient;i++){
        for (int j = 0;j<station->client[i].NbVoiture;j++){
            if(min>station->client[i].voiture[j].Pourcentage){
                min = station->client[i].voiture[j].Pourcentage;
                indMinV = j;
                indMinC = i;
            }
        }
    }
    printf("La voiture la moins chargee est %s %s du client %d\n",
        station->client[indMinC].voiture[indMinV].Marque,station->client[indMinC].voiture[indMinV].Model,
        indMinC+1);
}
```

Votre choix : 6

La voiture la plus chargee est Renault Clio du client 2

## 6.voiture plus chargée:

```
void PlusChargeeVoiture(STATION*station){
    float max = 0;
    int indMaxC= 0,indMaxV = 0;

    for (int i = 0;i<station->NbClient;i++){
        for (int j = 0;j<station->client[i].NbVoiture;j++){
            if(max<station->client[i].voiture[j].Pourcentage){
                max = station->client[i].voiture[j].Pourcentage;
                indMaxV = j;
                indMaxC = i;
            }
        }
    }
    printf("La voiture la plus chargee est %s %s du client %d\n",
    station->client[indMaxC].voiture[indMaxV].Marque,station->client[indMaxC].voiture[indMaxV].Model,
    indMaxC+1);
}
```

Votre choix : 5

La voiture la plus chargee est Tesla ModelX du client 1

## 7. Marque de la voiture la plus fréquente:

```
void MarqueDeVoiture(STATION*station) {
    char marque[50];
    int DejaProccede = 0;
    const int n = station->NbClient;
    char MatDejaFais[n*50][50];
    for (int i = 0; i < station->NbClient; i++) {
        for (int j = 0; j < station->client[i].NbVoiture; j++) {
            int DejaFais = 0;
            strcpy(marque, station->client[i].voiture[j].Marque);
            for (int p = 0; p < DejaProccede; p++) {
                if (strcmp(MatDejaFais[p], marque) == 0) {
                    DejaFais = 1;
                    break;
                }
            }
            if (DejaFais) {
                continue;
            }
            strcpy(MatDejaFais[DejaProccede], marque);
            DejaProccede++;
            int x = 0;
            for (int k = 0; k < station->NbClient; k++) {
                for (int z = 0; z < station->client[k].NbVoiture; z++) {
                    if (strcmp(station->client[k].voiture[z].Marque, marque) == 0) {
                        x++;
                    }
                }
            }

            printf("La marque %s apparait %d fois.\n", marque, x);
        }
    }
}

float Somme(STATION* station, int i) {
    float s = 0;
    for (int j = 0; j < station->client[i].NbPaielement; j++) {
        s += station->client[i].paielement[j].prix;
    }
    return s;
}
```

Votre choix : 7

La marque Tesla apparait 2 fois.

La marque Renault apparait 1 fois.

La marque Ferrari apparait 1 fois.

# Modifications:

## 1.Modifier client:

```
void ModifierClient(STATION*station){
    int ind;
    printf("Saisire le code client : ");
    scanf("%d",&ind);
    ind--;
    if (ind < 0 || ind >= station->NbClient) {
        printf("Code client invalide!\n");
        return;
    }
    printf("Saisir le nouveau nombre de voitures pour le client: ");
    scanf("%d", &station->client[ind].NbVoiture);
    printf("Saisir le nouveau nombre de paiements pour le client: ");
    scanf("%d", &station->client[ind].NbPaiement);
    station->client[ind].voiture = (VOITURE*)realloc(station->client[ind].voiture, station->client[ind].NbVoiture*sizeof(VOITURE));
    for (int i = 0; i < station->client[ind].NbVoiture; i++) {
        printf("Nouveau numero de serie (l'ancien est : %d): ",station->client[ind].voiture[i].NbSerie);
        scanf("%d", &station->client[ind].voiture[i].NbSerie);
        printf("Nouveau Marque (l'ancien est : %99s)", station->client[ind].voiture[i].Marque);
        scanf("%99s", station->client[ind].voiture[i].Marque);
        printf("Nouveau Model (l'ancien est : %99s)",station->client[ind].voiture[i].Model);
        scanf("%99s", station->client[ind].voiture[i].Model);
        printf("Nouveau Pourcentage (l'ancien est : %d)",station->client[ind].voiture[i].Pourcentage);
        scanf("%f", &station->client[ind].voiture[i].Pourcentage);
        printf("Nouveau TypeChargeur (l'ancien est : %d)",station->client[ind].voiture[i].TypeChargeur);
        scanf("%f", &station->client[ind].voiture[i].TypeChargeur);
    }
    for (int i = 0; i < station->client[ind].NbPaiement; i++) {
        printf("Saisire la Nouvelle date (l'ancien est : %d/%d/%d) : \n",station->client[ind].paiement[i].date.jour,
            station->client[ind].paiement[i].date.mois,station->client[ind].paiement[i].date.annee);
        printf("Jour : ");
        scanf("%d",&station->client[ind].paiement[i].date.jour);
        printf("Mois : ");
        scanf("%d",&station->client[ind].paiement[i].date.mois);
        printf("Annee : ");
        scanf("%d",&station->client[ind].paiement[i].date.annee);
        printf("Nouveaux prix (l'ancien est : %.2f): ",station->client[ind].paiement[i].prix);
        scanf("%f",&station->client[ind].paiement[i].prix);
    }
    printf("Client Modifie avec succes!!");
}
```

\*-----Modifier un client-----\*

Saisire le code client : 1  
Saisir le nouveau nombre de voitures pour le client: 1  
Saisir le nouveau nombre de paiements pour le client: 1  
Nouveau numero de serie (l'ancien est : 123): 321  
Nouveau Marque (l'ancien est :  
Nouveau Model (l'ancien est :  
Nouveau Pourcentage (l'ancien est : 1073741824)33.4  
Nouveau TypeChargeur (l'ancien est : 1)4  
Saisire la Nouvelle date (l'ancien est : 16/1/2003) :  
Jour : 16  
Mois : 01  
Annee : 2003  
Nouveaux prix (l'ancien est : 0.00): 500  
Client Modifie avec succes!!

Tesla)Tesla  
ModelX)Model3

## 2.Ajout client:

```
void AjoutClient(STATION*station){  
    station->client = (CLIENT*)realloc(station->client,((station->NbClient)+1)*sizeof(CLIENT));  
    RemplireClient(&station,station->NbClient);  
    station->NbClient +=1;  
}
```

\*-----Ajout un client-----\*

Tapez le Nombre des voiture : 1

Tapez le Nombre des paiements : 1

Tapez le numero du serie du voiture : 13567

Tapez la marque du serie du voiture : Lamborghini

Tapez le model du serie du voiture : Aventador

Tapez le pourcentage du serie du voiture : 21

Saisire le type du chargeur (1 = american , 2 = francais , 3 = anglais , 4 = african) :3

Tapez la date du paiements :

Jour : 1

Mois : 12

Annee : 2024

Tapez le montant : 301

### 3. Supprimer client:

```
void SupprimerClient(STATION*station){
    int ind;
    printf("Saisir le code du client a supprimer : ");
    scanf("%d", &ind);
    ind -= 1;
    if (ind < 0 || ind >= station->NbClient) {
        printf("Code client invalide!\n");
        return;
    }
    for (int i=ind;i <station->NbClient -1;i++) {
        station->client[i]=station->client[i+1];
    }
    station->client = (CLIENT*)realloc(station->client,(station->NbClient-1)*sizeof(CLIENT));
    station->NbClient -= 1;
    printf("Client supprime avec succes\n");
}
```

```
*-----supprimer un client-----*
Saisir le code du client a supprimer : 1
Client supprime avec succes
```

# 1.Modifier chargeur:

```
void ModifierChargeur(STATION*station){
    int ind;
    printf("Saisire le code chargeur : ");
    scanf("%d",&ind);
    ind-=1;
    if (ind < 0 || ind >= station->NbChargeur) {
        printf("Code chargeur invalide!\n");
        return;
    }
    printf("Donner la nouvelle etat d'utilisation (ancienne est : %d) :",station->chargeur[ind].EtatUtilisation);
    scanf("%d",&station->chargeur[ind].EtatUtilisation);
    printf("Donner Nouveaux TypeChargeur (ancienne est : %d) :",station->chargeur[ind].TypeChargeur);
    scanf("%d",&station->chargeur[ind].TypeChargeur);

    printf("Chargeur modifiee avec succes!\n");
}
```

\*-----Modification du Chargeur-----\*

Saisire le code chargeur : 1

Donner la nouvelle etat d'utilisation (ancienne est : 1) :2

Donner Nouveaux TypeChargeur (ancienne est : 1) :4

Chargeur modifiee avec succes!

## 2.Ajout chargeur:

```
void AjoutChargeur(STATION*station){  
    station->chargeur = (CHARGEUR*)realloc(station->chargeur,((station->NbChargeur)+1)*sizeof(CHARGEUR));  
    RemplireChargeur(&station,station->NbChargeur);  
    station->NbChargeur +=1;  
}
```

\*-----Ajout un Chargeur-----\*

Saisie l'etat d'utilisation du chargeur (1 = available , 2 = not availabe) : 1

Saisire le type du chargeur (1 = american , 2 = francais , 3 = anglais , 4 = african) :3



### 3. Supprimer chargeur:

```
void SupprimerChargeur(STATION*station) {
    int ind;
    printf("Saisir le code du chargeur a supprimer : ");
    scanf("%d", &ind);
    ind -= 1;
    if (ind < 0 || ind >= station->NbChargeur) {
        printf("Code chargeur invalide!\n");
        return;
    }
    for (int i = ind; i < station->NbChargeur - 1; i++) {
        station->chargeur[i] = station->chargeur[i+1];
    }

    station->chargeur = (CHARGEUR*)realloc(station->chargeur, (station->NbChargeur-1)*sizeof(CHARGEUR));
    station->NbChargeur -= 1;

    printf("Chargeur supprime avec succes\n");
}
```

```
*-----Suppression du Chargeur-----*
Saisir le code du chargeur a supprimer : 2
Chargeur supprime avec succes
```

# 1.Modifier voiture:

```
void ModifierVoiture(STATION*station){
    int indC,indV;
    printf("Saisir le code du client a modifier : ");
    scanf("%d", &indC);
    indC -= 1;
    if (indC < 0 || indC >= station->NbClient) {
        printf("Code client invalide!\n");
        return;
    }
    printf("Saisir le code du voiture a modifier : ");
    scanf("%d", &indV);
    indV -= 1;
    if (indV < 0 || indV >= station->client[indC].NbVoiture) {
        printf("Code Voiture invalide!\n");
        return;
    }
    printf("Donner la nouvelle serie de la voiture (ancienne est : %d) : ",station->client[indC].voiture[indV].NbSerie);
    scanf("%d",&station->client[indC].voiture[indV].NbSerie);
    printf("Donner la nouvelle marque de la voiture (ancienne est : %d) : ",station->client[indC].voiture[indV].Marque);
    scanf("%99s",&station->client[indC].voiture[indV].Marque);
    printf("Donner le nouveaux model de la voiture (ancienne est : %d) : ",station->client[indC].voiture[indV].Model);
    scanf("%99s",&station->client[indC].voiture[indV].Model);
    printf("Donner la nouvelle pourcentage de la voiture (ancienne est : %d) : ",station->client[indC].voiture[indV].Pourcentage);
    scanf("%.2f",&station->client[indC].voiture[indV].Pourcentage);
    printf("Donner le nouveaux typechargeur de la voiture (ancienne est : %d) : ",station->client[indC].voiture[indV].TypeChargeur);
    scanf("%d",&station->client[indC].voiture[indV].TypeChargeur);

    printf("voiture modifiee avec succes!\n");
}
void ModifierPaieement(STATION*station){
```

## 2.Ajout voiture:

```
void AjoutVoiture(STATION*station){
    int ind;
    printf("Saisire le code client : ");
    scanf("%d",&ind);
    ind -=1;
    if (ind < 0 || ind >= station->NbClient) {
        printf("Code client invalide!\n");
        return;
    }
    station->client[ind].voiture = (VOITURE*)realloc(station->client[ind].voiture,
    [(station->client[ind].NbVoiture + 1) * sizeof(VOITURE)]);
    station->client[ind].NbVoiture +=1;
    int n = station->client[ind].NbVoiture-1;

    printf("Tapez le numero du serie du voiture : ");
    scanf("%d",&station->client[ind].voiture[n].NbSerie);
    //model
    printf("Tapez la marque du serie du voiture : ");
    scanf("%99s",&station->client[ind].voiture[n].Marque);
    //marque
    printf("Tapez le model du serie du voiture : ");
    scanf("%99s",&station->client[ind].voiture[n].Model);
    //pourcentage
    printf("Tapez le pourcentage du serie du voiture : ");
    scanf("%f",&station->client[ind].voiture[n].Pourcentage);
}
```

```
*-----Ajout du Voiture-----*
Saisire le code client : 1
Tapez le numero du serie du voiture : 123
Tapez la marque du serie du voiture : Konigsegg
Tapez le model du serie du voiture : Agera
Tapez le pourcentage du serie du voiture : 40
```

### 3. Supprimer voiture:

```
void SupprimerVoiture(STATION*station){
    int indC,indV;
    printf("Saisir le code du client a supprimer : ");
    scanf("%d", &indC);
    indC -= 1;
    if (indC < 0 || indC >= station->NbClient) {
        printf("Code client invalide!\n");
        return;
    }
    printf("Saisir le code du voiture a supprimer : ");
    scanf("%d", &indV);
    indV -= 1;
    if (indV < 0 || indV >= station->client[indC].NbVoiture) {
        printf("Code Voiture invalide!\n");
        return;
    }
    for (int i=indV;i <station->client[i].NbVoiture -1;i++) {
        station->client[indC].voiture[i]=station->client[indC].voiture[i+1];
    }
    station->client[indC].voiture = (VOITURE*)realloc(station->client[indC].voiture,
    [(station->client[indC].NbVoiture-1)*sizeof(VOITURE)]);
    station->client[indC].NbVoiture -= 1;
    printf("Voiture supprime avec succes\n");
}
```

```
*-----Suppression du Voiture-----*
Saisir le code du client a supprimer : 1
Saisir le code du voiture a supprimer : 2
Voiture supprime avec succes
```

# 1.Modifier paiement:

```
void ModifierPaielement(STATION*station){
    int indC,indP;
    printf("Saisir le code du client a supprimer : ");
    scanf("%d", &indC);
    indC -= 1;
    if (indC < 0 || indC >= station->NbClient) {
        printf("Code client invalide!\n");
        return;
    }
    printf("Saisir le code du paiement a supprimer : ");
    scanf("%d", &indP);
    indP -= 1;
    if (indP < 0 || indP >= station->client[indC].NbPaielement) {
        printf("Code Paiement invalide!\n");
        return;
    }
    printf("Donner Nouvelle Date : (ancienne est : %d/%d/%d) : \n",station->client[indC].paielement[indP].date.jour,
station->client[indC].paielement[indP].date.mois,station->client[indC].paielement[indP].date.annee);
    printf("Jour : ");
    scanf("%d",&station->client[indC].paielement[indP].date.jour);
    printf("Mois : ");
    scanf("%d",&station->client[indC].paielement[indP].date.mois);
    printf("Annee : ");
    scanf("%d",&station->client[indC].paielement[indP].date.annee);
    printf("Donner Nouveaux prix : (ancien est : %.2f) : \n",station->client[indC].paielement[indP].prix);
    scanf("%f",&station->client[indC].paielement[indP].prix);
}
```

```
*-----Modification du Paiement-----*
Saisir le code du client a supprimer : 1
Saisir le code du paiement a supprimer : 1
Donner Nouvelle Date : (ancienne est : 4/3/1999)
Jour : 12
Mois : 05
Annee : 2023
Donner Nouveaux prix : (ancien est : 0.00) :
444
```

## 2.Ajout paiement:

```
void AjoutPaiement(STATION*station){
    int ind;
    printf("Saisire le code client : ");
    scanf("%d",&ind);
    ind -=1;
    if (ind < 0 || ind >= station->NbClient) {
        printf("Code client invalide!\n");
        return;
    }
    station->client[ind].paiement = (PAIEMENT*)realloc(station->client[ind].paiement,
    (station->client[ind].NbPaiement + 1) * sizeof(PAIEMENT));
    station->client[ind].NbPaiement +=1;
    int n = station->client[ind].NbPaiement-1;
    printf("Tapez la date du paiements : \n");
    printf("Jour : ");
    scanf("%d",&station->client[ind].paiement[n].date.jour);
    printf("Mois : ");
    scanf("%d",&station->client[ind].paiement[n].date.mois);
    printf("Annee : ");
    scanf("%d",&station->client[ind].paiement[n].date.annee);
    printf("Tapez le montant : ");
    scanf("%f",&station->client[ind].paiement[n].prix);
}
```

You, 17 hours ago • Traitement...

\*-----Ajout du Paiement-----\*

Saisire le code client : 1

Tapez la date du paiements :

Jour : 3

Mois : 7

Annee : 2024

Tapez le montant : 456

### 3. Supprimer paiement:

```
void SupprimerPaielement(STATION*station){
    int indC,indP;
    printf("Saisir le code du client a supprimer : ");
    scanf("%d", &indC);
    indC -= 1;
    if (indC < 0 || indC >= station->NbClient) {
        printf("Code client invalide!\n");
        return;
    }
    printf("Saisir le code du paiement a supprimer : ");
    scanf("%d", &indP);
    indP -= 1;
    if (indP < 0 || indP >= station->client[indC].NbPaielement) {
        printf("Code Paiement invalide!\n");
        return;
    }
    for (int i=indP;i <station->client[i].NbPaielement -1;i++) {
        station->client[indC].paiement[i]=station->client[indC].paiement[i+1];
    }
    station->client[indC].paiement = (PAIELEMENT*)realloc(station->client[indC].paiement,
    (station->client[indC].NbPaielement-1)*sizeof(PAIELEMENT));
    station->client[indC].NbPaielement -= 1;
    printf("Paiement supprime avec succes\n");
}
```

```
*-----Suppression du Paiement-----*
Saisir le code du client a supprimer : 1
Saisir le code du paiement a supprimer : 2
Paiement supprime avec succes
```

# Resultat:

```
RESULTAT* MeilleurClient(STATION* station) {
    RESULTAT* resultat = (RESULTAT*)malloc(3 * sizeof(RESULTAT));

    int indTop1 = -1, indTop2 = -1, indTop3 = -1;
    float max1 = 0.0, max2 = 0.0, max3 = 0.0;

    for (int i = 0; i < station->NbClient; i++) {
        float sommePaielements = Somme(station, i);

        if (sommePaielements > max1) {
            max3 = max2;
            indTop3 = indTop2;

            max2 = max1;
            indTop2 = indTop1;

            max1 = sommePaielements;
            indTop1 = i;
        } else if (sommePaielements > max2) {
            max3 = max2;
            indTop3 = indTop2;

            max2 = sommePaielements;
            indTop2 = i;
        } else if (sommePaielements > max3) {
            max3 = sommePaielements;
            indTop3 = i;
        }
    }

    resultat[0].CodeClient = indTop1;
    resultat[0].SommePaielements = max1;

    resultat[1].CodeClient = indTop2;
    resultat[1].SommePaielements = max2;

    resultat[2].CodeClient = indTop3;
    resultat[2].SommePaielements = max3;

    return resultat;
}
```



55 2 11 11 15 1 1 15



Format Digitale du Projet