ASSIGNMENT 7

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
struct car_rent
{
    char car_types[20];
    int price;
    struct car_rent *link;
};
struct car_rent *create(struct car_rent *start);
void display (struct car_rent *start);
struct car_rent *insert(struct car_rent *start);
struct car_rent *del(struct car_rent *start,char car_name[]);
struct car_rent *sort (struct car_rent *start);
void *search(struct car_rent *start,char car_name[]);
struct car_rent *modify(struct car_rent *start,char car_name[]);
struct car_rent *front=NULL;
struct car_rent *rear=NULL;
void create_queue();
void del_queue();
void display_queue();

int i,n;
void main()
{   char car_name[20];
    int ch;
    struct car_rent *start=NULL;
    while(1)
    {
    printf("\nEnter choice-\n 1) to create\n 2) to create using queue\n 3) to display\n 4) to display using queue\n 5) to insert\n 6) to delete\n 7) to delete using queue\n 8) to sort by price\n 9) to search\n 10) to modify:");
    scanf("%d",&ch);
    switch (ch)
    {
    case 1:
        start=create(start);
        break;
    case 2:
        create_queue();
        break;
    case 3:
        display(start);
        break;
    case 4:
        display_queue();
        break;
    case 5:
        start=insert(start);
        break;
    case 6:
        printf("Enter car_name you want to delete:");
        scanf("%s",car_name);
        start=del(start,car_name);
        break;
    case 7:
        del_queue();
        break;
    case 8:
        sort(start);
        break;
    case 9:
        printf("Enter car_name you want to search:");
        scanf("%s",car_name);
        search(start,car_name);
        break;
    case 10:
        printf("Enter car_name you want to modify:");
        scanf("%s",car_name);
        start=modify(start,car_name);
        break;
    default:
        printf("error!");

    }
}}
struct car_rent *create(struct car_rent* start)
{
    struct car_rent *p,*temp;
    printf("Enter no of type of car_names:");
    scanf("%d",&n);
    temp=start;
    for(i=0;i<n;i++)
     { temp=(struct car_rent*)malloc(sizeof(struct car_rent)*1);
       printf("enter car_types name:");
       scanf("%s",(temp->car_types));
       printf("enter price:");
       scanf("%d",&(temp->price));


    if(start==NULL)
    {
     start=temp;
     temp->link=NULL;

    }

  else
  {
        p=start;
        while(p->link!=NULL)
        p=p->link;
        p->link=temp;
        temp->link=NULL;
  }
}
return start;
}
void display (struct car_rent *start)
{
   struct car_rent *p;
   if(start==NULL)
   {
       printf("\n*Order List is Empty*\n\n");
       return;
   }
  p=start;
  printf("\ncar_name\tPrice\n");
  while(p!=NULL)

    {
        printf("\n%s\t %d\n",p->car_types,p->price);
        p=p->link;

    }
    printf("\n");
}
struct car_rent *insert(struct car_rent *start)
{
  int pos;
  struct car_rent *p,*temp;
  temp=(struct car_rent*)malloc(sizeof(struct car_rent)*1);
  printf("Enter location you want to insert:");
  scanf("%d",&pos);
    if(pos==1)
  {
    printf("enter car_types name:");
    scanf("%s",(temp->car_types));
    printf("enter price:");
    scanf("%d",&(temp->price));
    temp->link=start;
    start=temp;
    return start;
  }
     p=start;
     for(i=1;i<pos-1 && p!=NULL;i++)
        p=p->link;
     if(p==NULL)
        printf("\n*less no of car_names*\n\n");

    else
    {
        printf("enter car_types name:");
        scanf("%s",(temp->car_types));
        printf("enter price:");
        scanf("%d",&(temp->price));
        temp->link=p->link;
         p->link=temp;
    }
   return start;
}
struct car_rent *del(struct car_rent *start,char car_name[])
{
  struct car_rent *temp,*p;
  if(strcmp(start->car_types,car_name)==0)
  {
      temp=start;
      start=temp->link;
      free(temp);
      return start;
  }
      p=start;
      while(p->link!=NULL)
      { if(strcmp(p->link->car_types,car_name)==0)
          {
            temp=p->link;
            p->link=temp->link;
            free(temp);
            return start;
      }
     p=p->link;
     }
     }
struct car_rent *sort (struct car_rent *start)
{   struct car_rent *p,*q,*temp;
    temp=(struct car_rent*)malloc(sizeof(struct car_rent)*1);
   for(p=start;p->link!=NULL;p=p->link)
    {
        for(q=p->link;q!=NULL;q=q->link)
        {
        if(p->price > q->price)
          {
             temp->price=p->price;
             p->price=q->price;
             q->price=temp->price;
             strcpy(temp->car_types,p->car_types);
             strcpy(p->car_types,q->car_types);
             strcpy(q->car_types,temp->car_types);

          }
       }
    }
}
void *search(struct car_rent *start,char car_name[])
{
  struct car_rent *p;
  int flag=0;
      if(strcmp(start->car_types,car_name)==0)
         {
       printf("\nCAR_NAME FOUND\n");
       printf("\ncar_name\tPrice");
       printf("\n%s\t %d\n",start->car_types,start->price);
       flag++;
         }
       p=start;
       while(p->link!=NULL)
        {
            if(strcmp(p->link->car_types,car_name)==0)
          {
           printf("\nCAR_NAME FOUND\n");
           printf("\ncar_name\tPrice");
           printf("\n%s\t %d\n",p->link->car_types,p->link->price);
          }
           p=p->link;
           flag++;
        }
        if(flag==0)
        printf("\n\nnot found\n\n");
}
struct car_rent *modify(struct car_rent *start,char car_name[])
{
         struct car_rent *p;
    int choice;
    char ans1,ans2;
    int flag=0,chw=0;
      if(strcmp(start->car_types,car_name)==0)
     {
       printf("\nCAR_NAME FOUND\n");
       printf("\ncar_name\tPrice");
       printf("\n%s\t %d\n",start->car_types,start->price);
     do
     {
       printf("enter:\n 1) to modify car_types name\n 2) to modify car_types price:");
       scanf("%d",&choice);
       switch (choice)
       {
       case 1:
        printf("enter new car_types name:");
        scanf("%s",(start->car_types));
        break;
       case 2:
        printf("enter new car_types price:");
        scanf("%d",&(start->price));
        break;

       }

       printf("press 1 to continue modifying:");
       scanf("%d",&chw);
      }while(chw==1);

            flag++;
            return start;
      }

       p=start;
       while(p->link!=NULL)
         {
            if(strcmp(p->link->car_types,car_name)==0)
          {
           printf("\nCAR_NAME FOUND\n");
           printf("\ncar_name\tPrice");
           printf("\n%s\t %d\n",p->link->car_types,p->link->price);
            do
            {
            printf("enter:\n 1) to modify car_types name\n 2) to modify car_types price:");
            scanf("%d",&choice);

          switch (choice)
          {
           case 1:
           printf("enter new car_types name:");
           scanf("%s",(p->link->car_types));
           break;
           case 2:
           printf("enter new wire_types price:");
           scanf("%d",&(p->link->price));
           break;

           }printf("press 1 to continue modifying:");
            scanf("%d",&chw);
            }while(chw==1);


           }
               p=p->link;
               flag++;
               return start;
        }

        if(flag==0)
        printf("\n\nnot found\n\n");
}

void create_queue()
{
    struct car_rent *temp,*p;
     int n;
     printf("Enter Number of types of orders:");
     scanf("%d",&n);
     for(int i=0;i<n;i++)
     {

    if (rear==NULL)
    {
        rear=(struct car_rent*)malloc(sizeof(struct car_rent));
        printf("Enter car_types Name:");
        scanf("%s",(rear->car_types));
        printf("Enter car_types Price:");
        scanf("%d",&(rear->price));
        rear->link=NULL;
        front=rear;


    }

    else

    {
        temp=(struct car_rent*)malloc(sizeof(struct car_rent));
        printf("Enter car_types Name:");
        scanf("%s",(temp->car_types));
        printf("Enter car_types Price:");
        scanf("%d",&(temp->price));
        rear->link=temp;
        rear=temp;
    }
     }
}

void display_queue()
{
    struct car_rent *p;

    if(front==NULL)
    {
        printf("\n*Queue Underflow*\n");
    }
    else
    {    p=front;
        printf("\ncar_name\tPrice\n\n");
        while(p!=NULL)
           {
            printf("%s\t%d",p->car_types,p->price);
            printf("\n");
            p=p->link;
           }
    }
}
void del_queue()
{
    struct elec *temp;
    if(front==NULL)
    {
        printf("\n*Queue Underflow*\n");
    }
    else
    {
    temp=front;
    front=front->link;
    free (temp);
    }
}

/*
OUTPUT
Enter choice-
 1) to create
 2) to create using queue
 3) to display
 4) to display using queue
 5) to insert
 6) to delete
 7) to delete using queue
 8) to sort by price
 9) to search
 10) to modify:1
Enter no of type of car_names:1
enter car_types name:we
enter price:223

Enter choice-
 1) to create
 2) to create using queue
 3) to display
 4) to display using queue
 5) to insert
 6) to delete
 7) to delete using queue
 8) to sort by price
 9) to search
 10) to modify:2
Enter Number of types of orders:1
Enter car_types Name:er
Enter car_types Price:444

Enter choice-
 1) to create
 2) to create using queue
 3) to display
 4) to display using queue
 5) to insert
 6) to delete
 7) to delete using queue
 8) to sort by price
 9) to search
 10) to modify:3
```

```

car_name        Price

we       223


Enter choice-
 1) to create
 2) to create using queue
 3) to display
 4) to display using queue
 5) to insert
 6) to delete
 7) to delete using queue
 8) to sort by price
 9) to search
 10) to modify:4

car_name        Price

er      444

Enter choice-
 1) to create
 2) to create using queue
 3) to display
 4) to display using queue
 5) to insert
 6) to delete
 7) to delete using queue
 8) to sort by price
 9) to search
 10) to modify:5
Enter location you want to insert:1
enter car_types name:de
enter price:56

Enter choice-
 1) to create
 2) to create using queue
 3) to display
 4) to display using queue
 5) to insert
 6) to delete
 7) to delete using queue
 8) to sort by price
 9) to search
 10) to modify:3

car_name        Price

de       56

we       223


Enter choice-
 1) to create
 2) to create using queue
 3) to display
 4) to display using queue
 5) to insert
 6) to delete
 7) to delete using queue
 8) to sort by price
 9) to search
 10) to modify:4

car_name        Price

er      444

Enter choice-
 1) to create
 2) to create using queue
 3) to display
 4) to display using queue
 5) to insert
 6) to delete
 7) to delete using queue
 8) to sort by price
 9) to search
 10) to modify:6
Enter car_name you want to delete:de

Enter choice-
 1) to create
 2) to create using queue
 3) to display
 4) to display using queue
 5) to insert
 6) to delete
 7) to delete using queue
 8) to sort by price
 9) to search
 10) to modify:3

car_name        Price

we       223


Enter choice-
 1) to create
 2) to create using queue
 3) to display
 4) to display using queue
 5) to insert
 6) to delete
 7) to delete using queue
 8) to sort by price
 9) to search
 10) to modify:8

Enter choice-
 1) to create
 2) to create using queue
 3) to display
 4) to display using queue
 5) to insert
 6) to delete
 7) to delete using queue
 8) to sort by price
 9) to search
 10) to modify:9
Enter car_name you want to search:we

CAR_NAME FOUND

car_name        Price
we       223

Enter choice-
 1) to create
 2) to create using queue
 3) to display
 4) to display using queue
 5) to insert
 6) to delete
 7) to delete using queue
 8) to sort by price
 9) to search
 10) to modify:10
Enter car_name you want to modify:we

CAR_NAME FOUND

car_name        Price
we       223
enter:
 1) to modify car_types name
 2) to modify car_types price:1
enter new car_types name:dd
press 1 to continue modifying:1
enter:
 1) to modify car_types name
 2) to modify car_types price:2
enter new car_types price:45
press 1 to continue modifying:3

Enter choice-
 1) to create
 2) to create using queue
 3) to display
 4) to display using queue
 5) to insert
 6) to delete
 7) to delete using queue
 8) to sort by price
 9) to search
 10) to modify:3

car_name        Price

dd       45


Enter choice-
 1) to create
 2) to create using queue
 3) to display
 4) to display using queue
 5) to insert
 6) to delete
 7) to delete using queue
 8) to sort by price
 9) to search
 10) to modify:
*/
```