```c
// In this program we have implemented "Linked List" to perform

#include<stdio.h> //This header file provides basic I/O operations for program
#include<stdlib.h> //This header file includes functions involving memory allocation, process control, conversions and others.
                   //This header file includes string functions.

struct node //This is the structure which provides a template for node of Linked List. It shows a representation of a food item in menu.
    // It contains different attributes of node such as it's food name, qauntity, price, data, pointer to next node and pointer to previous node.

#include<string.h>
    //We are using one to perform various task of admin as well as customer using one single structre, we have created one single struct
    //catering to purpose of menu item as well as customer's order.
{   char   foodname[50];
    int quantity;
    float price;
    int data;
    struct node *next;


};


//global struct pointers which are used throughout the program to create linked list and maintain it.
struct node *headc = NULL,*newnode,*tailc = NULL;
struct   node *heada = NULL, *taila = NULL;
struct   node *head_s;

//This function prints the options available for admin to choose
void adminmenu()
{
    printf("\n\t\t\t\t\t\t\t1. View total sales\n");
    printf("\t\t\t\t\t\t\t2. Add new items in the order menu\n");
    printf("\t\t\t\t\t\t\t3. Delete items from the order menu\n");
    printf("\t\t\t\t\t\t\t4. Display order menu\n");
    printf("\t\t\t\t\t\t\t5. Back To Main Menu \n\n");
    printf("\t\t\t\t\t\t\t   Enter Your Choice --->");
```

```
}

//This function prints the options available for customer to choose
void customermenu()
{
    printf("\n\t\t\t\t\t\t\t1. Place your order\n");
    printf("\t\t\t\t\t\t\t2. View your ordered items\n");
    printf("\t\t\t\t\t\t\t3. Delete an item from order\n");
    printf("\t\t\t\t\t\t\t4. Display final bill\n");
    printf("\t\t\t\t\t\t\t5. Back To Main Menu \n\n");
    printf("\t\t\t\t\t\t\t   Enter Your Choice --->");
}

//This function creates a node for admin's Linked List
struct node* createadmin(struct node *head,int data, char foodname[25
], float price)
{
    newnode = (struct node*)malloc(sizeof(struct node));

    newnode->data = data;
    newnode->price = price;
    newnode-> quantity = 0;
    strcpy(newnode->foodname,foodname);

    newnode->next = NULL;

    struct node *temp = head;

    if(temp==NULL)
        heada = taila = newnode;
    else
    {
        while(temp->next!=NULL)
            temp=temp->next;

        temp->next=newnode;
        taila = newnode;
    }

    return heada;
}

//This function creates a node for customer's Linked List
struct node* createcustomer(struct node *head,int data,int quantity)
{
    newnode = (struct node*)malloc(sizeof(struct node));

    struct node *temp1 = heada;
    int flag = 0;
    while(temp1!=NULL)
    {
```

```
        if(temp1->data==data)
        {
            flag = 1;
            break;
        }
        temp1 = temp1->next;
    }

    if(flag==1)
    {
        newnode->data = data;
        newnode->price = quantity*(temp1->price);
        newnode-> quantity = quantity;
        strcpy(newnode->foodname,temp1->foodname);
        newnode->next = NULL;
        struct node *temp = head;

        if(temp==NULL)
            headc = tailc = newnode;
        else
        {
            while(temp->next!=NULL)
                temp=temp->next;

            temp->next=newnode;
            tailc = newnode;
        }


    }
    else
    {
        printf("\n\t\t\t\t\t\t\tThis item is not present in the menu!\
n");
    }
    return headc;
}

//This function displays the respective entire Linked List whose head
pointer is passed to it
void displayList(struct node *head)
{
    struct node *temp1 = head;
    if(temp1==NULL)
    {
        printf("\n\t\t\t\t\t\t\t\tList is empty!!\n\n");
    }
    else
    {
        printf("\n");
```

```
        while(temp1!=NULL)
        {
            if(temp1->quantity==0)
                printf("\t\t\t\t\t\t\t%d\t%s\t\t%0.2f\n",temp1->data,
temp1->foodname,temp1->price);
            else
            {
                printf("\t\t\t\t\t\t\t%d\t%s\t\t%d\t\t%0.2f\n",temp1->
data,temp1->foodname,temp1->quantity,temp1->price);
            }

            temp1 = temp1->next;
        }
        printf("\n");
    }

}

//This function maintains the total value of sales done by maintaining
 another Linked List which keeps a track of
//Total sales made to each customer represented by each node
struct node* totalsales(int data,int quantity)
{
    newnode = (struct node*)malloc(sizeof(struct node));
    int flag = 0;

           node *temp1 = heada;
    struct
    while(temp1->data!=data)
    {
        temp1 = temp1->next;
    }

    newnode->data = data;
    newnode->price = quantity*(temp1->price);
    newnode-> quantity = quantity;
    strcpy(newnode->foodname,temp1->foodname);
    newnode->next = NULL;

    struct node *temp = head_s;

    if(temp==NULL)
        head_s = newnode;
    else
    {
        while(temp->next!=NULL)
        {
            if(temp->data==data)
            {
                flag = 1;
                break;
```

```
            }
            temp=temp->next;
        }

        if(flag==1)
        {
            temp->quantity += newnode-> quantity;
            temp->price += newnode->price;
        }
        else
        {
            temp->next=newnode;
        }
    }

    return head_s;
}

//This function performs task of calculating total sales for each
customer
void calculatetotsales()
{
            node *temp = headc;
    struct
    while(temp!=NULL)
    {
        head_s = totalsales(temp->data, temp->quantity);
        temp=temp->next;
    }
}

//This function performs the task of deleting the data from Linked
List whose respective head pointer is passed.
//Here, data to be deleted is passed as a parameter.
struct node* delete(int data,struct node *head, struct node* tail)
{
    if(head==NULL)
    {
        printf("\n\t\t\t\t\t\t\tList is empty\n");
    }
    else
    {
        struct node* temp;
        if(data==head->data)
        {
            temp = head;
            head = head->next;
            free(temp);
        }
        else if(data==tail->data)

        {
```

```
            temp = tail;
            tail->next = NULL;
            free(temp);
        }
        else
        {
            temp = head;
            struct node* temp1=temp;
            while(data!=temp->data)
            {
                temp1=temp;
                temp = temp->next;
            }
            temp1->next = temp->next;
            free(temp);
        }
    }
    return head;
}

//This function performs the task of deleting food item from admin's
Linked List.
int deleteadmin()
{
    printf("\n\t\t\t\t\tEnter serial no. of the food item which is to
be deleted: ");
    int num;
    scanf("%d",&num);

           node* temp=heada;
    struct while(temp!=NULL)
    {
           (temp->data == num)
        if{
            heada = delete(num, heada, taila);
            return 1;
        }
        temp=temp->next;
    }

    return 0;
}

//This function performs the task of deleting food item from customer'
s Linked List i.e. customer's ordered food item
int deletecustomer()
{
    printf("\n\t\t\t\t\tEnter serial no. of the food item which is to
be deleted: ");
    int num;
```

```
    scanf("%d",&num);

    struct node* temp=headc;
    while(temp!=NULL)
    {
        if (temp->data == num)
        {
            headc = delete(num, headc, tailc);
            return 1;
        }
        temp=temp->next;
    }

    return 0;
}

//This function displays the total bill of food items ordered by
customer.
void displaybill()
{
    displayList(headc);
    struct node *temp = headc;
    float total_price = 0;
    while (temp!=NULL)
    {
        total_price +=(temp->quantity)*(temp->price);
        temp = temp->next;
    }

    printf("\t\t\t\t\t\t\tTotal price: %0.02f\n",total_price);

}

//This function performs the task of deleting entire Linked List.
struct node* deleteList(struct node* head)
{
    if(head==NULL)
    {
        return NULL;
    }
    else
    {
        struct node* n, *temp=head;

        while(temp!=NULL)
        {
            n=temp->next;
            free(temp);
            temp=n;

        }
```

```
        head=NULL;
    }

    return head;
}

//This function opens up the admin section and provides it's interface
 and functionalities
void admin()
{
    printf("\n\t\t\t\t\t\t
   ------------------------------------------------\n");
    printf("\t\t\t\t\t\t\t    ADMIN SECTION\n");
    printf("\t\t\t\t\t\t
   ------------------------------------------------------\n");
    while(1)
    {
        adminmenu(); //prints admin functionlity list

        int opt;
        scanf("%d",&opt); //scans choice of user

        if(opt==5)
            break;

        switch (opt) //switch-case block which executes according to
the option selected by user
        {
            case 1:
                displayList(head_s);
                break;
            case 2:

                printf("\n\t\t\t\t\t\t\tEnter serial no. of the food
item: ");
                int num,flag = 0;
                char name[50];
                float price;
                scanf("%d",&num);

                struct node *temp = heada;

                while(temp!=NULL)
                {
                    if(temp->data==num)
                    {
                        printf("\n\t\t\t\t\t\tFood item with given
serial number already exists!!\n\n");
                        flag = 1;
                        break;
```

```
                    }
                    temp = temp->next;
                }

                if(flag==1)
                    break;

                printf("\t\t\t\t\t\t\tEnter food item name: ");
                scanf("%s",name);
                printf("\t\t\t\t\t\t\tEnter price: ");
                scanf("%f",&price);
                heada = createadmin(heada, num, name, price);
                printf("\n\t\t\t\t\t\t\tNew food item added to the
list!!\n\n");
                break;
            case 3:
                if(deleteadmin())
                {
                    printf("\n\t\t\t\t\t\t### Updated list of food
items menu ###\n");
                    displayList(heada);
                }
                else
                    printf("\n\t\t\t\t\t\tFood item with given serial
number doesn't exist!\n\n");

                break;
            case 4:
                printf("\n\t\t\t\t\t\t   ### Order menu ###\n");
                displayList(heada);
                break;

            default:
                printf("\n\t\t\t\t\t\tWrong Input !! PLease choose
valid option\n");
                break;
        }
    }
}

//This function opens up the customer section and provides it's
interface and functionalities
void customer()
{
    int flag=0,j=1;
    char ch;
    printf("\n\t\t\t\t\t
   ------------------------------------------\n");
    printf("\t\t\t\t\t\t    CUSTOMER SECTION\n");
    printf("\t\t\t\t\t
```

```
    ----------------------------------------------\n");
     while(1)
     {
         customermenu(); //prints customer functionlity list

         int opt;
         scanf("%d",&opt); //scans choice of user

         if(opt==5)
             break;

         //switch-case block which executes according to the option
selected by user
         switch (opt)
         {
             case 1:
                 displayList(heada);
                 printf("\n\t\t\t\t\t\tEnter number corresponding to
the item you want to order: ");
                 int n;
                 scanf("%d",&n);
                 printf("\t\t\t\t\t\tEnter quantity: ");
                 int quantity;
                 scanf("%d",&quantity);
                 headc = createcustomer(headc, n, quantity);
                 break;
             case 2:
                 printf("\n\t\t\t\t\t\t  ### List of ordered items
 ###\n");
                 displayList(headc);
                 break;
             case 3:
                 if(deletecustomer())
                 {
                     printf("\n\t\t\t\t\t\t### Updated list of your
ordered food items ###\n");
                     displayList(headc);
                 }
                 else
                     printf("\n\t\t\t\t\t\tFood item with given serial
number doesn't exist!!\n");
                 break;
             case 4:
                 calculatetotsales();
                 printf("\n\t\t\t\t\t\t
                 displaybill();             ### Final Bill ###\n");
                 headc = deleteList(headc);
                 printf("\n\t\t\t\t\t\tPress any key to return to main
menu:\n\t\t\t\t\t\t");
                 fflush(stdin);
```

```
                ch=fgetc(stdin);
                flag=1;
                break;

            default:
                printf("\n\t\t\t\t\t\tWrong Input !! PLease choose
valid option\n");
                break;
        }
        if(flag==1)
            break;
    }
}

//This function prints the welcome interface and opens the main menu
where you can select the option where you want to go.
void mainnenu()
{
    printf("\n
                                        ***********************************
***********************************\n");
    printf("
WELCOME TO RESTAURANT MANAGEMENT SYSTEM\n");
    printf(
"                                         ***********************************
***********************************\n\n\n");
    printf("\t\t\t\t\t\t\t1. ADMIN SECTION--> \n");
    printf("\t\t\t\t\t\t\t2. CUSTOMER SECTION--> \n");
    printf("\t\t\t\t\t\t\t3. Exit--> \n\n");
    printf("\t\t\t\t\t\t\tEnter Your Choice --->");
}

int main() //From here the actual program execution begins
{
    //Here we have initialized admin's Linked List i.e. Food Menu with
 5 items
    heada = createadmin(heada,1,"Hot and Sour Soup",100);
    heada = createadmin(heada,2,"Manchow Soup",200);
    heada = createadmin(heada,3,"Manchurian Noodles",150);
    heada = createadmin(heada,4,"Fried Rice",180);
    heada = createadmin(heada,5,"Hakka Noodles",80);

    while(1)
    {
        mainnenu();
        int choice;
        scanf("%d",&choice); //scans choice of user

        if(choice==3)
        {
```

```
            printf("\n\n\t\t\t\t\t\t\t**********Thank you!!**********\
n");
            break;
        }

        //switch-case block which executes according to the option
selected by user
        switch (choice)
        {
            case 1:
                admin();
                break;
            case 2:
                customer();
                break;
            case 3:
                break;

            default:
                printf("\n\t\t\t\t\t\tWrong Input !! PLease choose
valid option\n");
                break;
        }
    }
}
```