**Medical Store Management System**

Introduction

The Medical Store Management System is a software application designed to manage the operations of a medical store. It provides functionality for both administrators and customers to interact with the system. The system allows the addition, modification, and deletion of medicines in the store, as well as the generation of bills for customers.

**Features**

**1. Administrator Section**

The administrator section provides the following features:

a. Login

The administrator needs to log in with a valid username and password to access the system.

b. Medicine Management

Add Medicine: Add new medicines to the store with details such as medicine ID, name, quantity, and price.

Insert Medicine at Specific Location: Insert a medicine at a specific position in the list.

Sort Medicines: Sort medicines by either medicine ID or name.

Delete Medicine: Delete medicines from the store.

c. Search Medicines

Search for medicines based on either medicine ID or name.

d. Display Medicines

Display the list of medicines in both sorted and unsorted ways.

e. Backup

Provide options to import and export the database, creating a backup of the data.

2. **Customer Section**

The customer section allows customers to interact with the system and make purchases. Features include:

a. Add Medicine to Purchase List

Customers can add medicines to their purchase list by specifying the medicine ID and quantity.

b. Delete Medicine from Purchase List

Customers can remove medicines from their purchase list.

c. Search Medicines

Search for medicines in the store based on either medicine ID or name.

d. Generate Bill

Customers can generate a bill for their purchase, displaying the total price and details of each medicine.

e. View Purchase List

Customers can view their current purchase list.

**Implementation Details**

The system is implemented in C programming language and utilizes file handling for storing and retrieving data. It includes functions for reading and writing data to files, as well as sorting and searching functionalities for efficient data management.

**Code**

#include <stdio.h>

#include <stdlib.h>

#include<windows.h>

struct node

{

    int id;

    char med\_name[50];

    int quantity;

    float price;

    struct node \*next;

};

struct node \*head=NULL,\*last=NULL,\*p,obj,\*q=NULL;

struct node \*headc=NULL,\*head\_s=NULL,\*newnode,\*tailc=NULL;

int found=0,i,n;

void login()

{

    char AdminID[]="Siddharth", password[]="123456", id[20], p[20];

    int n=1, x, y;

    printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*If you are admin kindly login\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

    do

    {

        printf("\nAdmin\_ID:");

        scanf("%s", &id);

        fflush(stdout);

        printf("\nPassword:");

        scanf("%s", &p);

        fflush(stdout);

        x=strcmp(id, AdminID);

        y=strcmp(p, password);

        if(x==0 && y==0)

        {

            printf("\nSucessfully Logged In");

            admin();

            break;

        }

        else

        {

            printf("\nWrong Password, try again", 3-n);

            getch();

            n++;

        }

        if(n>5)

        {

            printf("\nAccess Denied");

            getch();

        }

    }

    while (n<=3);

}

//display medicines

void displayMed()

{

    readdatabase();

    p=head;

    if(p==NULL)

    {

    printf("list is empty");

    }

    else

{

    printf("\n-------------------------------------------------\n");

    printf("Medicine No|| Name || Quantity || Price ||");

    printf("\n--------------------------------------------------\n   ");

    for(i=0;p!=NULL;i++)

    {

     printf("\n%d\t %s\t\t %d\t\t %f",(p->id),p->med\_name,(p->quantity),p->price);

     p=p->next;

    }}}

//reading the Medical database

void readdatabase()

    {

    int i,filempty=0;

    FILE \*infile=fopen("Medicine\_database.txt","r");

    p=head;

    fseek(infile,0,SEEK\_END);

    if(p==NULL)

    {

    p=(struct node\*)malloc(sizeof(struct node));

    head=p;

    }

    int len=(int)ftell(infile);

    if(len<=0)

    {

    filempty=1;

    printf("File empty");

    p=NULL;

    head=p;

    }

    if(filempty==0)

    {

    rewind(infile);

    while(fscanf(infile,"\n%d\t\t %s\t\t %d\t\t %f",&(p->id),p->med\_name,&(p->quantity),&(p->price)))

{

    if(feof(infile))

    {

    break;

    }

    p->next=(struct node\*)malloc(sizeof(struct node));

    p=p->next;

    last=p;

    p->next=NULL;

}}}

//writing into Medical database

void writedatabase()

    {

    FILE \*ne=fopen("Medicine\_database.txt","w");

    int i;

    struct node \*temp;

    temp=head;

    if(temp==NULL)

    {

        printf("Empty File");

    }

    else

    {

    for(i=0;temp!=NULL;i++)

    {

    fprintf(ne,"\n%d\t\t %s\t\t %d\t\t %f",(temp->id),temp->med\_name,(temp->quantity),temp->price);

   temp=temp->next;

    }//for

    }//else

    fclose(ne);

    }

//inserting the medicine at head or in the end

void insertMed()

{   do{

    readdatabase();

    found=0;

    p=head;

    printf("\n-------------------------------------------------\n");

    printf("Medicine No|| Name || Quantity || Price ||");

    printf("\n--------------------------------------------------\n   ");

    scanf("\t%d %s %d %f",&(obj.id),obj.med\_name,&(obj.quantity),&(obj.price));

                            for(i=1;p!=NULL;i++)

                            {

                            if(p->id==obj.id)

                            {

    printf("\n%d\t %s\t\t %d\t\t %f",(p->id),p->med\_name,(p->quantity),p->price);

                            found=1;

                            printf("\n\t\t Oopps  !!!  Medicine Already Present ....\n") ;

                            }

                            p=p->next;

                            }}while(found==1);

    struct node \*p;

    int i=0;

    p=last;

    if(last==NULL)

    {

    i=1;

    last=(struct node\*)malloc(sizeof(struct node));

    last->next=NULL;

    last->id=obj.id;

    strcpy(last->med\_name,obj.med\_name);

    last->quantity=obj.quantity;

    last->price=obj.price;

    p=last;

    head=last;

    }//if

else{i=0;

    p=last;

    p->next=(struct node\*)malloc(sizeof(struct node));

    p=p->next;

    last=p;

    p->id=obj.id;

    strcpy(p->med\_name,obj.med\_name);

    p->quantity=obj.quantity;

    p->price=obj.price;

    p->next=NULL;

    }//i for

   writedatabase();

   printf("\n\n\t\tMedicine Added Successfully ....!!!!!!!\n\n") ;

}

//inserting the medicine at the specific location

struct node\* insertbw()

{

    readdatabase();

    int i,loc;

    int id,quantity;

    char med\_name[50];

    float price;

    printf("\nEnter the position:");

    scanf("%d",&loc);

   do{

        readdatabase();

    printf("\n-------------------------------------------------\n");

    printf("Medicine No|| Name || Quantity || Price ||");

    printf("\n--------------------------------------------------\n   ");

    scanf("\t%d %s %d %f",&(id),med\_name,&(quantity),&(price));

    found=0;

    p=head;

    for(i=1;p!=NULL;i++)

                            {

                            if(p->id==id)

                            {

    printf("\n%d\t %s\t\t %d\t\t %f",(p->id),p->med\_name,(p->quantity),p->price);

                            found=1;

                            printf("\n\t\t Oopps  !!!  Medicine Already Present ....\n") ;

                            }

                            p=p->next;

                            }}while(found==1);

    p=head;

    q=head;

    p=(struct node\*)malloc(sizeof(struct node));

    p->id=id;

    strcpy(p->med\_name,med\_name);

    p->quantity=quantity;

    p->price=price;

    p->next=NULL;

    if(loc==1)

    {

        p->next=head;

        head=p;

        writedatabase();

        return(p);

    }

    q=head;

    for(i=1;i<loc-1;i++)

    {

        if(q!=NULL)

        {

            q=q->next;

        }

    }

    p->next=q->next;

    q->next=p;

    n=n+1;

    writedatabase();

                             printf("\n\n\t\tMedicine Added Successfully ....!!!!!!!\n\n") ;

    return(head);

}

//reading the sorted database

void readsorted()

    {

    int i,filempty=0;

    FILE \*infile=fopen("SortedMed\_database.txt","r");

    p=head;

    fseek(infile,0,SEEK\_END);

    if(p==NULL)

    {

    p=(struct node\*)malloc(sizeof(struct node));

    head=p;

    }

    int len=(int)ftell(infile);

    if(len<=0)

    {

    filempty=1;

    printf("File empty");

    p=NULL;

    head=p;

    }

    if(filempty==0)

    {

    rewind(infile);

    while(fscanf(infile,"\n%d\t\t %s\t\t %d\t\t %f",&(p->id),p->med\_name,&(p->quantity),&(p->price)))

{

    if(feof(infile))

    {

    break;

    }

    p->next=(struct node\*)malloc(sizeof(struct node));

    p=p->next;

    last=p;

    p->next=NULL;

}}}

//writing the sorted medicines

void writesorted()

    {

    FILE \*ne=fopen("SortedMed\_database.txt","w");

    int i;

    struct node \*temp;

    temp=head;

    if(temp==NULL)

    {

        printf("Empty File");

    }

    else

    {

    for(i=0;temp!=NULL;i++)

    {

    fprintf(ne,"\n%d\t\t %s\t\t %d\t\t %f",(temp->id),temp->med\_name,(temp->quantity),temp->price);

   temp=temp->next;

    }

    }

    fclose(ne);

    }

//sorting the medicines by their medicine id

void sortMed()

    {

    int k;

    readdatabase();

    int temp;

    struct node \*i,\*j;

    for(i=head;i!=NULL;i=i->next)

    {

        for(j=i->next;j!=NULL;j=j->next)

        {

            if((i->id)>(j->id))

            {

                obj.id=i->id;

                obj.quantity=i->quantity;

                obj.price=i->price;

                i->id=j->id;

                i->quantity=j->quantity;

                i->price=j->price;

                j->id=obj.id;

                j->quantity=obj.quantity;

                j->price=obj.price;

                strcpy(obj.med\_name,i->med\_name);

                strcpy(i->med\_name,j->med\_name);

                strcpy(j->med\_name,obj.med\_name);

             }}}//i for

    writesorted();

    }//void sort

//display the sorted medicines

void displaysortedMed()

{

    readsorted();

    int i;

    p=head;

    if(p==NULL)

    {

    printf("list is empty");

    }

    else

        {

    printf("\n-------------------------------------------------\n");

    printf("Medicine No|| Name || Quantity || Price ||");

    printf("\n--------------------------------------------------\n   ");

    for(i=0;p!=NULL;i++)

    {

    printf("\n%d\t %s\t\t %d\t\t %f",(p->id),p->med\_name,(p->quantity),p->price);

    p=p->next;

}}}

//deleting the medicines

void deleteMed()

{   struct node \*p,\*q;

    readsorted();

    int loc,i;

    displayMed();

    printf("\nEnter the location to delete:\t");

    scanf("%d",&loc);

    if(p!=NULL)

    {

        if(loc==1)

    {

        p=head;

        head=head->next;

        writesorted();

        displaysortedMed();

        free(p);

        return(head);

    }

    else{q=head;}

    for(i=1;i<loc-1;i++)

    {

        q=q->next;

    }

    p=q->next;

    q->next=p->next;

    writesorted();

    displaysortedMed();

    free(p);

    n=n-1;

    }

    else

    {

    printf("\nSorry,The list is empty");

    }}

//searching the medicines

void SearchMed()

    {

        readsorted();

    int id;

    char med\_name[50];

    p=head;

    int choice2;

                do{

                printf("\n--------------------------------------------------------\n");

                printf("                       SEARCH MENU                      ");

                printf("\n--------------------------------------------------------\n\n");

                printf("\t1. Medicine Id \n");

                printf("\t2. Medicine Name\n");

                printf("\t0. EXIT\n");

                printf("\n--------------------------------------------------------\n");

                printf("Choice:\t");

                scanf("%d",&choice2);

                switch(choice2)

                {

                    case 1:

                            readsorted();

                                p=head;

                            printf("\n\nEnter the Medicine No:\t");

                            scanf("%d",&id);

    printf("\n-------------------------------------------------\n");

    printf("Medicine No|| Name || Quantity || Price ||");

    printf("\n--------------------------------------------------\n   ");

                           for(i=1;p!=NULL;i++)

                            {

                            if(p->id==id)

                            {

     printf("\n%d\t %s\t\t %d\t\t %f",(p->id),p->med\_name,(p->quantity),p->price);

                            found=1;

                            }

                            p=p->next;

                            }

                            if(!found)

                            {

                            printf("\n\t Oops !!!! Medicine not found\n\n");

                            }

                            printf("\n\n\n0. MAIN MENU\t\t\t");

                            printf("1. SUB MENU\n\n");

                            printf("Choice:\t");

                            scanf("%d",&choice2);

                            break;

                    case 2:

                            readsorted();

                                p=head;

                            printf("\n\nEnter the Medicine Name:\t");

                            scanf("%s",med\_name);

    printf("\n-------------------------------------------------\n");

    printf("Medicine No|| Name || Quantity || Price ||");

    printf("\n--------------------------------------------------\n   ");

                            for(i=1;p!=NULL;i++)

                            {

                            if(strcmp(p->med\_name,med\_name)==0)

                            {

    printf("\n%d\t %s\t\t %d\t\t %f",(p->id),p->med\_name,(p->quantity),p->price);

                            found=1;

                            }

                            p=p->next;

                            }

                            if(!found)

                            {

                            printf("\n\t Oops !!!! Medicine with this name not found\n\n");

                            }

                            printf("\n\n\n0. MAIN MENU\t\t\t");

                            printf("1. SUB MENU\n\n");

                            printf("Choice:\t");

                            scanf("%d",&choice2);

                            break;

                }

                }while(choice2!=0);

        }

//displaying medicines in both sorted and unsorted way

void SpecialMedDis()

{

    int choice2;

     do{

                printf("\n--------------------------------------------------------\n");

                printf("                    DISPLAY MENU                      ");

                printf("\n--------------------------------------------------------\n\n");

                printf("\t1. Unsorted Medicines \n");

                printf("\t2. Sorted Medicines\n");

                printf("\t0. EXIT\n");

                printf("\n--------------------------------------------------------\n");

                printf("Choice:\t");

                scanf("%d",&choice2);

                switch(choice2)

                {

                    case 1: system("CLS");

                            readdatabase();

                             p=head;

                            if(p==NULL)

                            {

                            printf("list is empty");

                            }

                            else

                                {

                            printf("\n-------------------------------------------------\n");

                            printf("Medicine No|| Name || Quantity || Price ||");

                            printf("\n--------------------------------------------------\n   ");

                            for(i=0;p!=NULL;i++)

                            {

                             printf("\n%d\t %s\t\t %d\t\t %f",(p->id),p->med\_name,(p->quantity),p->price);

                            p=p->next;

                            }}

                            printf("\n\n\n0. MAIN MENU\t\t\t");

                            printf("1. SUB MENU\n\n");

                            printf("Choice:\t");

                            scanf("%d",&choice2);

                            break;

                    case 2: system("CLS");

                            readsorted();

                            p=head;

                            if(p==NULL)

                            {

                            printf("list is empty");

                            }

                            else

                                {

                            printf("\n-------------------------------------------------\n");

                            printf("Medicine No|| Name || Quantity || Price ||");

                            printf("\n--------------------------------------------------\n   ");

                            for(i=0;p!=NULL;i++)

                            {

                             printf("\n%d\t %s\t\t %d\t\t %f",(p->id),p->med\_name,(p->quantity),p->price);

                            p=p->next;

                            }}

                            printf("\n\n\n0. MAIN MENU\t\t\t");

                            printf("1. SUB MENU\n\n");

                            printf("Choice:\t");

                            scanf("%d",&choice2);

                            break;

                }

                }while(choice2!=0);

}

//keeping backup of the data

void backup()

{

    int choice2;

                do{

                printf("\n--------------------------------------------------------\n");

                printf("                        MENU                      ");

                printf("\n--------------------------------------------------------\n\n");

                printf("\t1. Import Database \n");

                printf("\t2. Export Database\n");

                printf("\t0. EXIT\n");

                printf("\n--------------------------------------------------------\n");

                printf("Choice:\t");

                scanf("%d",&choice2);

                switch(choice2)

                {

                    case 1:

                            import();

                            printf("\n\t IMPORTING BACKUP \n");

                            printf("\n\t IMPORTING BACKUP \n");

                            printf("\n\t---- Done---- Successfully Imported data !!!!!!.");

                            printf("\n\n\n0. MAIN MENU\t\t\t");

                            printf("1. SUB MENU\n\n");

                            printf("Choice:\t");

                            scanf("%d",&choice2);

                            break;

                    case 2:

                            export1();

                            printf("\n\n\n0. MAIN MENU\t\t\t");

                            printf("1. SUB MENU\n\n");

                            printf("Choice:\t");

                            scanf("%d",&choice2);

                            break;

                }

                }while(choice2!=0);

}

//exporting data to backup file

void export1()

{   readsorted();

    FILE \*xp=fopen("Backup.txt","w");

    int i;

    struct node \*temp;

    temp=head;

    if(temp==NULL)

    {

        printf("list is empty");

    }

    else

    {

    printf("\n\t TAKING BACKUP \n");

    printf("\n\t TAKING BACKUP\n");

    for(i=0;temp!=NULL;i++)

    {

        fprintf(xp,"\n%d\t\t %s\t\t %d\t\t %f",(temp->id),temp->med\_name,temp->quantity,(temp->price));

    temp=temp->next;

    }//for

    printf("\n\t---- Done---- BACKUP TAKEN !!!!!!.");

    }//else

    fclose(xp);

}

//importing data to backup file

void import()

{

    int i,filempty=0;

    FILE \*imp=fopen("Backup.txt","r");

    p=head;

    fseek(imp,0,SEEK\_END);

    if(p==NULL)

    {

    p=(struct node\*)malloc(sizeof(struct node));

    head=p;

    }

    int len=(int)ftell(imp);

    if(len<=0)

    {

    filempty=1;

    printf("File empty");

    p=NULL;

    head=p;

    }

    if(filempty==0)

    {

    rewind(imp);

    while(fscanf(imp,"\n%d\t\t %s\t\t %d\t\t %f",&(p->id),p->med\_name,p->quantity,&(p->price)))

    {

    if(feof(imp))

    {

    break;

    }

    p->next=(struct node\*)malloc(sizeof(struct node));

    p=p->next;

    last=p;

    p->next=NULL;

    }

    }

    writesorted();

}

//admin section

void admin()

{

    system("CLS");

    printf("\n                                 |-------------------------------------------------------------------------|\n");

    printf("                                 |                       WELCOME TO ADMIN SECTION                          |\n");

    printf("                                 |-------------------------------------------------------------------------|\n\n\n");

    int choice2;

    do{

    printf("<<<<-Press 1 for adding medicines->>>>\n");

    printf("<<<<-Press 2 for adding medicine at specific location->>>>\n");

    printf("<<<<-Press 3 for sorting the medicines by medicine id or name->>>>\n");

    printf("<<<<-Press 4 for deleting the medicines->>>>\n");

    printf("<<<<-Press 5 for searching the medicines->>>>\n");

    printf("<<<<-Press 6 for displaying the medicines->>>>\n");

    printf("<<<<-press 7 for keeping the backup of the data->>>>\n");

    printf("<<<<-Press 0 for Exit->>>>\n\n");

    printf("Enter your choice ");

    scanf("%d",&choice2);

    switch(choice2){

        case 1:

            insertMed();

            printf("\n\t\tPress Any Key to Continue ....!!!!!!!\n") ;

            getch();

            break;

        case 2:

            displayMed();

            insertbw();

            printf("\n\t\tPress Any Key to Continue ....!!!!!!!\n") ;

            getch();

            break;

        case 3:

            sortMed();

            printf("->>>>Medicines sorted successfully<<<<-\n");

            displaysortedMed();

            printf("\n\n\n0. MAIN MENU\t\t\t");

            printf("1. SUB MENU\n\n");

            printf("Choice:\t");

            scanf("%d",&choice2);

            break;

        case 4:

            deleteMed();

            printf("\n\n\n0. MAIN MENU\t\t\t");

            printf("1. SUB MENU\n\n");

            printf("Choice:\t");

            scanf("%d",&choice2);

            break;

        case 5:

            SearchMed();

            printf("\n\n\n0. MAIN MENU\t\t\t");

            printf("1. SUB MENU\n\n");

            printf("Choice:\t");

            scanf("%d",&choice2);

            break;

        case 6:

            SpecialMedDis();

            printf("\n\n\n0. MAIN MENU\t\t\t");

            printf("1. SUB MENU\n\n");

            printf("Choice:\t");

            scanf("%d",&choice2);

            break;

        case 7:

            backup();

            printf("\n\n\n0. MAIN MENU\t\t\t");

            printf("1. SUB MENU\n\n");

            printf("Choice:\t");

            scanf("%d",&choice2);

            break;

        default:

            break;

    }

    }while(choice2!=0);

}

//

void addMed()

{

    readdatabase();

    p=head;

    found=0;

    int id,quantity;

    printf("\n-------------------------\n");

    printf("Medicine No|| Quantity ||");

    printf("\n------------------------\n   ");

    scanf("\t%d %d",&id,&quantity);

    for(i=1;p!=NULL;i++)

    {

        if(p->id==id)

            {

                found=1;

                break;

            }

            p=p->next;

    }

    if(found==0)

        printf(">>>>Medicine not found->>>>\n");

    if(found==1)

    {

     if(quantity<p->quantity){

      newnode = (struct node\*)malloc(sizeof(struct node));

      newnode->id = id;

      newnode->price = (quantity)\*(p->price);

      newnode-> quantity = quantity;

      strcpy(newnode->med\_name,p->med\_name);

      newnode->next = NULL;

      struct node \*temp=headc;

      if(temp==NULL)

            headc = newnode;

        else

        {

            while(temp->next!=NULL)

                temp=temp->next;

            temp->next=newnode;

        }

       printf(">>>>Medicine successfully added to your purchase list->>>>\n");

    }

     else{

        printf(">>>>Oops Not enough quantity present\n");

        printf("\n-------------------------\n");

        printf("Medicine No|| Quantity ||");

        printf("\n------------------------\n   ");

        printf("\n%d\t %s\t\t %d\t\t %f",(p->id),p->med\_name,(p->quantity),p->price);

     }

    }

}

//display the list of medicines

void displayList(struct node \*head)

{

    printf("\t---------------------------------------------------------------------------------\n");

    printf("\t|    Medicine No.\tMedicine Name\t\tQuantity\t Total Price\t|");

    struct node \*temp1 = head;

    if(temp1==NULL)

    {

        printf("\n\t\t\t\t\tList is empty!!\n\n");

    }

    else

    {

        printf("\n");

        while(temp1!=NULL)

        {

            if(temp1->quantity==0)

                printf("\t|\t%d\t\t%s\t\t%0.2f\t|\n",temp1->id,temp1->med\_name,temp1->price);

            else

            {

                printf("\t|\t%d\t\t%s\t\t\t%d\t\t%0.2f\t|\n",temp1->id,temp1->med\_name,temp1->quantity,temp1->price);

            }

            temp1 = temp1->next;

        }

    }

    printf("\t|                                                                               |");

    printf("\n\t---------------------------------------------------------------------------------\n");

}

struct node\* delete(int id,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,\*q;

        if(id==head->id)

        {

            temp = head;

            if(head->next!=NULL)

                head=head->next;

            else

                head=NULL;

            free(temp);

        }

        else

        {

            temp = head;

            while(id!=temp->id)

            {

                q=temp;

                temp = temp->next;

            }

            q->next=temp->next;

            free(temp);

        }

    }

    return head;

}

int delCustMed()

{

    printf("\n\t\t\t\t\tEnter medicine id to be deleted: ");

    int num;

    scanf("%d",&num);

    struct node\* temp=headc;

    while(temp!=NULL)

    {

        if (temp->id == num)

        {

            headc = delete(num, headc, tailc);

            return 1;

        }

        temp=temp->next;

    }

    return 0;

}

//total bill generated

void displaybill()

{

    displayList(headc);

    struct node \*temp = headc;

    float total\_price = 0;

    while (temp!=NULL)

    {

        total\_price +=temp->price;

        temp = temp->next;

    }

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

}

//customer section

void customer()

{

    printf("\n                                 |-------------------------------------------------------------------------|\n");

    printf("                                 |                      WELCOME TO CUSTOMER SECTION                        |\n");

    printf("                                 |-------------------------------------------------------------------------|\n\n\n");

    system("CLS");

    int choice2;

    do{

    printf("<<<<-Press 1 for adding medicine to your purchase list->>>>\n");

    printf("<<<<-Press 2 for Deleting medicine from purchase list->>>>\n");

    printf("<<<<-Press 3 for Searching medicines by medicine id or name->>>>\n");

    printf("<<<<-Press 4 Generating the bill->>>>\n");

    printf("<<<<-Press 5 to view your purchase list\n");

    printf("<<<<-Press 0 for Exit->>>>\n\n");

    printf("Enter your choice ");

    scanf("%d",&choice2);

    switch(choice2){

        case 1:

            addMed();

            printf("\n\t\tPress Any Key to Continue ....!!!!!!!\n") ;

            getch();

            break;

        case 2:

            if(delCustMed())

                printf("<<<<-Medicine successfully deleted->>>>>\n");

            else

                printf("<<<<-Oops medicine with this id is not in your purchase list->>>>\n");

            printf("\n\t\tPress Any Key to Continue ....!!!!!!!\n") ;

            getch();

            break;

        case 3:

            SearchMed();

            printf("\n\n\n0. MAIN MENU\t\t\t");

            printf("1. SUB MENU\n\n");

            printf("Choice:\t");

            scanf("%d",&choice2);

            break;

        case 4:

            printf("\n\t\t\t\t\t\t\t<<<<-Final Bill->>>>\n");

            displaybill();

            printf("\n\n\n0. MAIN MENU\t\t\t");

            printf("1. SUB MENU\n\n");

            printf("Choice:\t");

            scanf("%d",&choice2);

            break;

        case 5:

            displayList(headc);

            printf("\n\t\tPress Any Key to Continue ....!!!!!!!\n") ;

            getch();

            break;

        default:

            break;

    }

    }while(choice2!=0);

}

void mainnenu()

{

    printf("\n                                 |-------------------------------------------------------------------------|\n");

    printf("                                 |                       WELCOME TO MAIN MENU                              |\n");

    printf("                                 |-------------------------------------------------------------------------|\n\n\n");

    printf("\n\n<<<<-Press 1 for Admin section->>>>\n");

    printf("<<<<-Press 2 for Customer section->>>>\n");

    printf("<<<<-Press 3 to exit->>>>\n");

    printf("<<<<- Your Choice Please->>>>");

}

int main()

{

    system("Color F4");

    int c;

    printf("\n\n\t\t\t\t\tXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX\n");

    printf("\t\t\t\t\t|                           MEDICAL STORE MANAGEMENT SYSTEM                       |\n");

    printf("\t\t\t\t\tXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX\n");;

    printf("\n\n\t=> This Project Is About Medical Store\n");

    printf("\t=> In This Project we Can Add Medicine ,Customer Details\n");

    printf("\t=> We Can Modified & Delete Existing Record\n");

    printf("\t=> We Can Also Search Medicine ,Customer Details\n");

    printf("\t=> It's Helpfull For Display Stock Of Medicine\n\n");

    do

    {

        printf("<<<<-Press 1 for main menu->>>>");

        c = (getche());

        switch (c)      //This 'SWITCH' structure will ask the user for input from [1] to [7] and will display error on Invalid Entry.

        {

        case '1':

            mainnenu();

            while(1)

            {

                int choice;

                scanf("%d",&choice);

                if(choice==3)

                {

                    break;

                }

                switch (choice)

                {

                case 1:

                    login();

                    goto br;

                case 2:

                     customer();

                    goto br;

                default:

                    printf("\n\t\t\t\t\t\tWrong Input !! PLease choose valid option\n");

                    break;

                }

            }

        default:

            printf("\n\t\t\t\t\t\tWrong Input !! PLease choose valid option\n");

            break;

        }

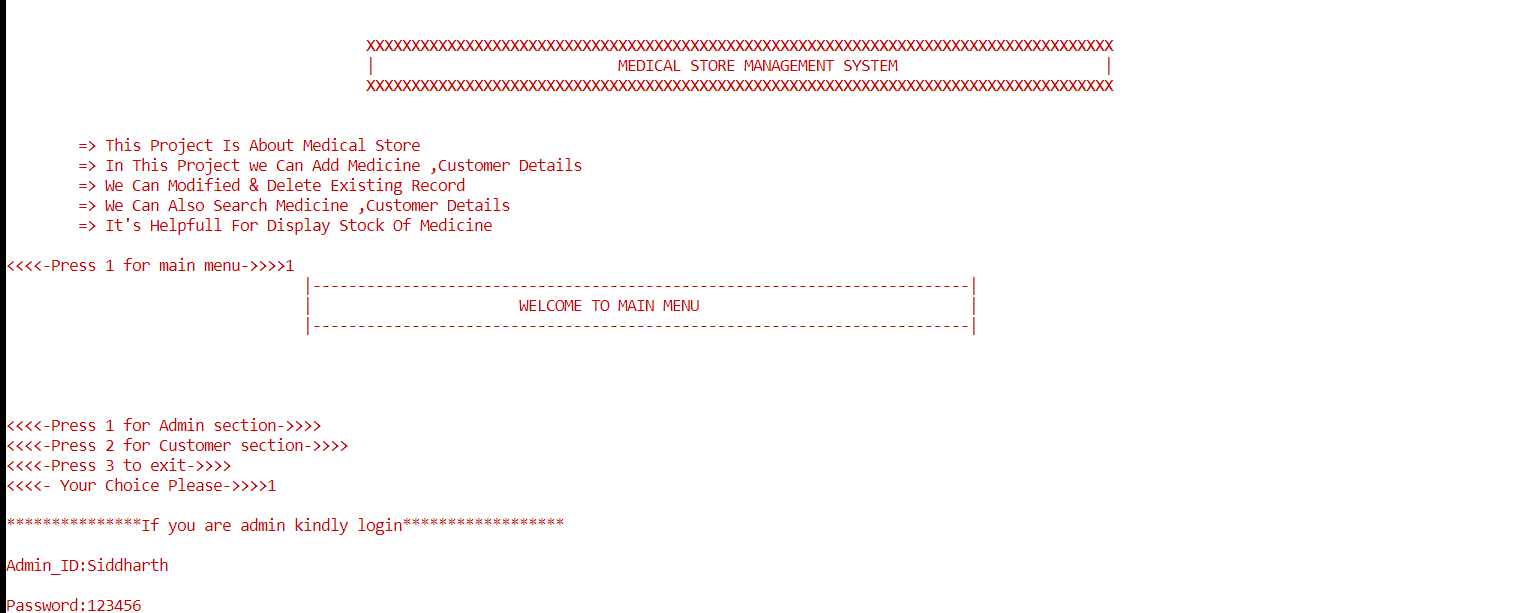
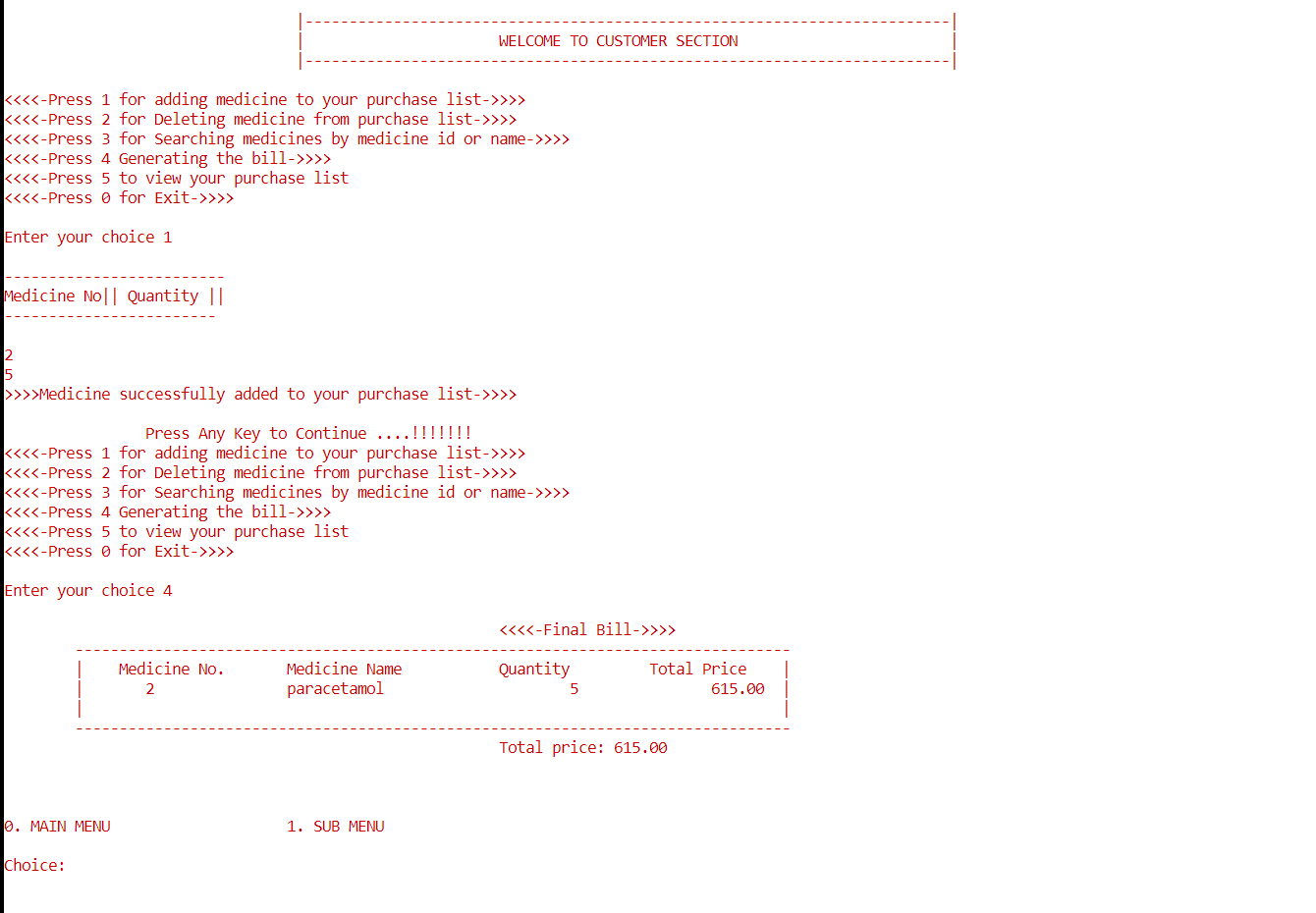
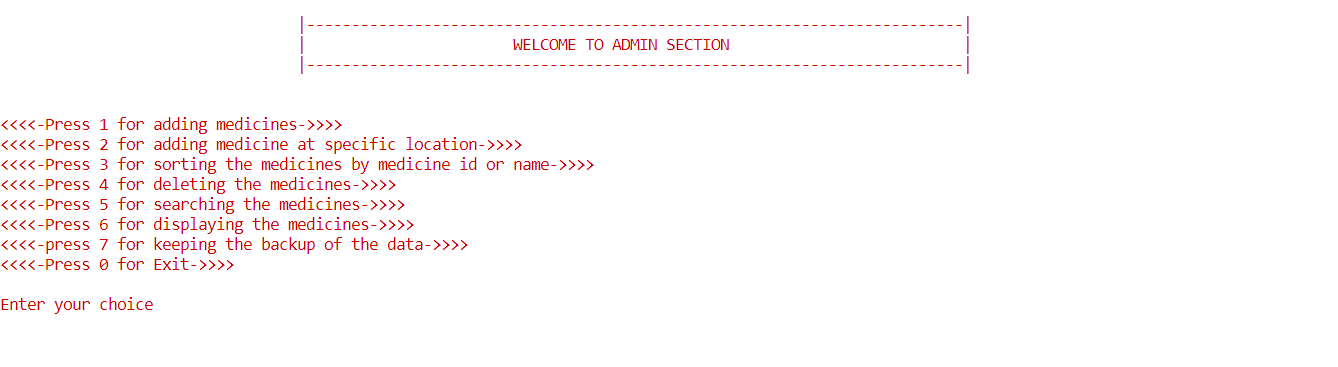
    }

    while(c != '1');

    br:

    return 0;

}

****