```c
#include<stdio.h>

#include<stdlib.h>

#include<conio.h>

#include<string.h>

int password();

void addrecord();

void viewrecord();

void editrecord();

void editpassword();

void deleterecord();

struct record

{

    char time[6];

    char name[30];

    char place[25];

    char duration[10];

    char note[500];

} ;

int main()

{

    int ch;

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

    printf("\t*Welcome to Diary Management System*\n");

    printf("\t**********************************");
```

```
    while(1)
    {
        printf("\n\n\t\tMAIN MENU:");
        printf("\n\n\tADD RECORD\t[1]");
        printf("\n\tVIEW RECORD\t[2]");
        printf("\n\tEDIT RECORD\t[3]");
        printf("\n\tDELETE RECORD\t[4]");
        printf("\n\tEDIT PASSWORD\t[5]");
        printf("\n\tEXIT\t\t[6]");
        printf("\n\n\tENTER YOUR CHOICE:");
        scanf("%d",&ch);
        switch(ch)
        {
        case 1:
            addrecord();
            break;
        case 2:
            viewrecord();
            break;
        case 3:
            editrecord();
            break;
        case 4:
            deleterecord();
```

```
            break;
        case 5:
            editpassword();
            break;
        case 6:
            printf("\n\n\t\tTHANK YOU FOR USING THE SOFTWARE ");
            getch();
            exit(0);
        default:
            printf("\nYOU ENTERED WRONG CHOICE..");
            printf("\nPRESS ANY KEY TO TRY AGAIN");
            getch();
            break;
        }
        system("cls");
    }
    return 0;
}
void addrecord( )
{
    system("cls");
    FILE *fp ;
    char another = 'Y' ,time[10];
    struct record e ;
```

```
    char filename[15];

    int choice;

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

    printf("\t\t* WELCOME TO THE ADD MENU *");

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

    printf("\n\n\tENTER DATE OF YOUR RECORD:[dd-mm-yyyy]:");

    fflush(stdin);

    gets(filename);

    fp = fopen (filename, "ab+" ) ;

    if ( fp == NULL )

    {

        fp=fopen(filename,"wb+");

        if(fp==NULL)

        {

            printf("\nSYSTEM ERROR...");

            printf("\nPRESS ANY KEY TO EXIT");

            getch();

            return ;

        }

    }

    while ( another == 'Y'|| another=='y' )

    {

        choice=0;

        fflush(stdin);
```

```
        printf ( "\n\tENTER TIME:[hh:mm]:");
        scanf("%s",time);
        rewind(fp);
        while(fread(&e,sizeof(e),1,fp)==1)
        {
            if(strcmp(e.time,time)==0)
            {
                printf("\n\tTHE RECORD ALREADY EXISTS.\n");
                choice=1;
            }
        }
        if(choice==0)
        {
            strcpy(e.time,time);
            printf("\tENTER NAME:");
            fflush(stdin);
            gets(e.name);
            fflush(stdin);
            printf("\tENTER PLACE:");
            gets(e.place);
            fflush(stdin);
            printf("\tENTER DURATION:");
            gets(e.duration);
            fflush(stdin);
```

```
            printf("\tNOTE:");
            gets(e.note);
            fwrite ( &e, sizeof ( e ), 1, fp ) ;
            printf("\nYOUR RECORD IS ADDED...\n");
        }
        printf ( "\n\tADD ANOTHER RECORD...(Y/N) " ) ;
        fflush ( stdin ) ;
        another = getchar( ) ;
    }
    fclose ( fp ) ;
    printf("\n\n\tPRESS ANY KEY TO EXIT...");
    getch();
}
void viewrecord( )
{
    FILE *fpte ;
    system("cls");
    struct record customer ;
    char time[6],choice,filename[14];
    int ch;
    printf("\n\n\t\t*****************************\n");
    printf("\t\t* HERE IS THE VIEWING MENU *");
    printf("\n\t\t*****************************\n\n");
    choice=password();
```

```
    if(choice!=0)

    {

        return ;

    }

    do

    {

        printf("\n\tENTER THE DATE OF RECORD TO BE VIEWED:[yyyy-mm-dd]:");

        fflush(stdin);

        gets(filename);

        fpte = fopen ( filename, "rb" ) ;

        if ( fpte == NULL )

        {

            puts ( "\nTHE RECORD DOES NOT EXIST...\n" ) ;

            printf("PRESS ANY KEY TO EXIT...");

            getch();

            return ;

        }

        system("cls");

        printf("\n\tHOW WOULD YOU LIKE TO VIEW:\n");

        printf("\n\t1.WHOLE RECORD OF THE DAY.");

        printf("\n\t2.RECORD OF FIX TIME.");

        printf("\n\t\tENTER YOUR CHOICE:");

        scanf("%d",&ch);

        switch(ch)
```

```
        {
        case 1:
            printf("\nTHE WHOLE RECORD FOR %s IS:",filename);
            while ( fread ( &customer, sizeof ( customer ), 1, fpte ) == 1 )
            {
                printf("\n");
                printf("\nTIME: %s",customer.time);
                printf("\nMEETING WITH: %s",customer.name);
                printf("\nMEETING AT: %s",customer.place);
                printf("\nDURATION: %s",customer.duration);
                printf("\nNOTE: %s",customer.note);
                printf("\n");
            }
            break;
        case 2:
            fflush(stdin);
            printf("\nENTER TIME:[hh:mm]:");
            gets(time);
            while ( fread ( &customer, sizeof ( customer ), 1, fpte ) == 1 )
            {
                if(strcmp(customer.time,time)==0)
                {
                    printf("\nYOUR RECORD IS:");
                    printf("\nTIME: %s",customer.time);
```

```
                    printf("\nMEETING WITH: %s",customer.name);

                    printf("\nMEETING AT: %s",customer.place);

                    printf("\nDUARATION: %s",customer.duration);

                    printf("\nNOTE: %s",customer.note);

                }

            }

            break;

        default:
            printf("\nYOU TYPED SOMETHING ELSE...\n");

            break;

        }

        printf("\n\nWOULD YOU LIKE TO CONTINUE VIEWING...(Y/N):");

        fflush(stdin);

        scanf("%c",&choice);

    }
    while(choice=='Y'||choice=='y');

    fclose ( fpte ) ;

    return ;

}

void editrecord()

{

    system("cls");

    FILE *fpte ;

    struct record customer ;

    char time[6],choice,filename[14];
```

```
    int num,count=0;

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

    printf("\t\t* WELCOME TO THE EDITING MENU *");

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

    choice=password();

    if(choice!=0)

    {

        return ;

    }

    do

    {

        printf("\n\tENTER THE DATE OF RECORD TO BE EDITED:[yyyy-mm-dd]:");

        fflush(stdin);

        gets(filename);

        printf("\n\tENTER TIME:[hh:mm]:");

        gets(time);

        fpte = fopen ( filename, "rb+" ) ;

        if ( fpte == NULL )

        {

            printf( "\nRECORD DOES NOT EXISTS:" ) ;

            printf("\nPRESS ANY KEY TO GO BACK");

            getch();

            return;

        }
```

```
        while ( fread ( &customer, sizeof ( customer ), 1, fpte ) == 1 )
        {
            if(strcmp(customer.time,time)==0)
            {
                printf("\nYOUR OLD RECORD WAS AS:");
                printf("\nTIME: %s",customer.time);
                printf("\nMEETING WITH: %s",customer.name);
                printf("\nMEETING AT: %s",customer.place);
                printf("\nDURATION: %s",customer.duration);
                printf("\nNOTE: %s",customer.note);
                printf("\n\n\t\tWHAT WOULD YOU LIKE TO EDIT..");
                printf("\n1.TIME.");
                printf("\n2.MEETING PERSON.");
                printf("\n3.MEETING PLACE.");
                printf("\n4.DURATION.");
                printf("\n5.NOTE.");
                printf("\n6.WHOLE RECORD.");
                printf("\n7.GO BACK TO MAIN MENU.");
                do
                {
                    printf("\n\tENTER YOUR CHOICE:");
                    fflush(stdin);
                    scanf("%d",&num);
                    fflush(stdin);
```

```
                switch(num)

                {

                case 1:
                    printf("\nENTER THE NEW DATA:");

                    printf("\nNEW TIME:[hh:mm]:");

                    gets(customer.time);

                    break;

                case 2:
                    printf("\nENTER THE NEW DATA:");

                    printf("\nNEW MEETING PERSON:");

                    gets(customer.name);

                    break;

                case 3:
                    printf("\nENTER THE NEW DATA:");

                    printf("\nNEW MEETING PLACE:");

                    gets(customer.place);

                    break;

                case 4:
                    printf("\nENTER THE NEW DATA:");

                    printf("\nDURATION:");

                    gets(customer.duration);

                    break;

                case 5:
                    printf("ENTER THE NEW DATA:");

                    printf("\nNOTE:");

                    gets(customer.note);

                    break;
```

```
            case 6:
                printf("\nENTER THE NEW DATA:");

                printf("\nNEW TIME:[hh:mm]:");

                gets(customer.time);

                printf("\nNEW MEETING PERSON:");

                gets(customer.name);

                printf("\nNEW MEETING PLACE:");

                gets(customer.place);

                printf("\nDURATION:");

                gets(customer.duration);

                printf("\nNOTE:");

                gets(customer.note);

                break;

            case 7:
                printf("\nPRESS ANY KEY TO GO BACK...\n");

                getch();

                return ;

                break;

            default:
                printf("\nYOU TYPED SOMETHING ELSE...TRY AGAIN\n");

                break;

            }

        }
        while(num<1||num>8);

        fseek(fpte,-sizeof(customer),SEEK_CUR);

        fwrite(&customer,sizeof(customer),1,fpte);
```

```
            fseek(fpte,-sizeof(customer),SEEK_CUR);
            fread(&customer,sizeof(customer),1,fpte);
            choice=5;
            break;
        }
    }
    if(choice==5)
    {
        system("cls");
        printf("\n\t\tEDITING COMPLETED...\n");
        printf("--------------------\n");
        printf("THE NEW RECORD IS:\n");
        printf("--------------------\n");
        printf("\nTIME: %s",customer.time);
        printf("\nMEETING WITH: %s",customer.name);
        printf("\nMEETING AT: %s",customer.place);
        printf("\nDURATION: %s",customer.duration);
        printf("\nNOTE: %s",customer.note);
        fclose(fpte);
        printf("\n\n\tWOULD YOU LIKE TO EDIT ANOTHER RECORD.(Y/N)");
        scanf("%c",&choice);
        count++;
    }
    else
```

```
        {
            printf("\nTHE RECORD DOES NOT EXIST::\n");
            printf("\nWOULD YOU LIKE TO TRY AGAIN...(Y/N)");
            scanf("%c",&choice);
        }
    }
    while(choice=='Y'||choice=='y');
    fclose ( fpte ) ;
    if(count==1)
        printf("\n%d FILE IS EDITED...\n",count);
    else if(count>1)
        printf("\n%d FILES ARE EDITED..\n",count);
    else
        printf("\nNO FILES EDITED...\n");
    printf("\tPRESS ENTER TO EXIT EDITING MENU.");
    getch();
}
int password()
{
    char pass[15]= {0},check[15]= {0},ch;
    FILE *fpp;
    int i=0,j;
    printf("::FOR SECURITY PURPOSE::");
    printf("::ONLY THREE TRIALS ARE ALLOWED::");
```

```
for(j=0; j<3; j++)
{
    i=0;
    printf("\n\n\tENTER THE PASSWORD:");
    pass[0]=getch();
    while(pass[i]!='\r')
    {
        if(pass[i]=='\b')
        {
            i--;
            printf("\b");
            printf(" ");
            printf("\b");
            pass[i]=getch();
        }
        else
        {
            printf("*");
            i++;
            pass[i]=getch();
        }
    }
    pass[i]='\0';
    fpp=fopen("SE","r");
```

```
        if (fpp==NULL)
        {
            printf("\nERROR WITH THE SYSTEM FILE...[FILE MISSING]\n");
            getch();
            return 1;
        }
        else
            i=0;
        while(1)
        {
            ch=fgetc(fpp);
            if(ch==EOF)
            {
                check[i]='\0';
                break;
            }
            check[i]=ch-5;
            i++;
        }
        if(strcmp(pass,check)==0)
        {
            printf("\n\n\tACCESS GRANTED...\n");
            return 0;
        }
```

```
        else

        {

            printf("\n\n\tWRONG PASSWORD..\n\n\tACCESS DENIED...\n");

        }

    }

    printf("\n\n\t::YOU ENTERED WRONG PASSWORD::YOU ARE NOT ALLOWED TO ACCESS
ANY FILE::\n\n\tPRESS ANY KEY TO GO BACK...");

    getch();

    return 1;

}

void editpassword()

{

    system("cls");

    printf("\n");

    char pass[15]= {0},confirm[15]= {0},ch;

    int choice,i,check;

    FILE *fp;

    fp=fopen("SE","rb");

    if(fp==NULL)

    {

        fp=fopen("SE","wb");

        if(fp==NULL)

        {

            printf("SYSTEM ERROR...");

            getch();
```

```
            return ;

        }

        fclose(fp);

        printf("\nSYSTEM RESTORED...\nYOUR PASSWORD IS 'ENTER'\n PRESS ENTER T
O CHANGE PASSWORD\n\n");

        getch();

    }

    fclose(fp);

    check=password();

    if(check==1)

    {

        return ;

    }

    do

    {

        if(check==0)

        {

            i=0;

            choice=0;

            printf("\n\n\tENTER THE NEW PASSWORD:");

            fflush(stdin);

            pass[0]=getch();

            while(pass[i]!='\r')

            {
```

```
        if(pass[i]=='\b')
        {
            i--;
            printf("\b");
            printf(" ");
            printf("\b");
            pass[i]=getch();
        }
        else
        {
            printf("*");
            i++;
            pass[i]=getch();
        }
    }
    pass[i]='\0';
    i=0;
    printf("\n\tCONFIRM PASSWORD:");
    confirm[0]=getch();
    while(confirm[i]!='\r')
    {
        if(confirm[i]=='\b')
        {
            i--;
```

```
                printf("\b");
                printf(" ");
                printf("\b");
                confirm[i]=getch();
            }
            else
            {
                printf("*");
                i++;
                confirm[i]=getch();
            }
        }
        confirm[i]='\0';
        if(strcmp(pass,confirm)==0)
        {
            fp=fopen("SE","wb");
            if(fp==NULL)
            {
                printf("\n\t\tSYSTEM ERROR");
                getch();
                return ;
            }
            i=0;
            while(pass[i]!='\0')
```

```
                {
                    ch=pass[i];
                    putc(ch+5,fp);
                    i++;
                }
                putc(EOF,fp);
                fclose(fp);
            }
            else
            {
                printf("\n\tTHE NEW PASSWORD DOES NOT MATCH.");
                choice=1;
            }
        }
    }
    while(choice==1);
    printf("\n\n\tPASSWORD CHANGED...\n\n\tPRESS ANY KEY TO GO BACK...");
    getch();
}
void deleterecord( )
{
    system("cls");
    FILE *fp,*fptr ;
    struct record file ;
    char filename[15],another = 'Y' ,time[10];;
```

```
    int choice,check;
    printf("\n\n\t\t************************\n");
    printf("\t\t* WELCOME TO DELETE MENU*");
    printf("\n\t\t************************\n\n");
    check = password();
    if(check==1)
    {
        return ;
    }
    while ( another == 'Y' )
    {
        printf("\n\n\tHOW WOULD YOU LIKE TO DELETE.");
        printf("\n\n\t#DELETE WHOLE RECORD\t\t\t[1]");
        printf("\n\t#DELETE A PARTICULAR RECORD BY TIME\t[2]");
        do
        {
            printf("\n\t\tENTER YOU CHOICE:");
            scanf("%d",&choice);
            switch(choice)
            {
            case 1:
                printf("\n\tENTER THE DATE OF RECORD TO BE DELETED:[yyyy-mm-
dd]:");
                fflush(stdin);
```

```
            gets(filename);
            fp = fopen (filename, "wb" ) ;
            if ( fp == NULL )
            {
                printf("\nTHE FILE DOES NOT EXISTS");
                printf("\nPRESS ANY KEY TO GO BACK.");
                getch();
                return ;
            }
            fclose(fp);
            remove(filename);
            printf("\nDELETED SUCCESFULLY...");
            break;
        case 2:
            printf("\n\tENTER THE DATE OF RECORD:[yyyy-mm-dd]:");
            fflush(stdin);
            gets(filename);
            fp = fopen (filename, "rb" ) ;
            if ( fp == NULL )
            {
                printf("\nTHE FILE DOES NOT EXISTS");
                printf("\nPRESS ANY KEY TO GO BACK.");
                getch();
                return ;
```

```
            }
            fptr=fopen("temp","wb");
            if(fptr==NULL)
            {
                printf("\nSYSTEM ERROR");
                printf("\nPRESS ANY KEY TO GO BACK");
                getch();
                return ;
            }
            printf("\n\tENTER THE TIME OF RECORD TO BE DELETED:[hh:mm]:");
            fflush(stdin);
            gets(time);
            while(fread(&file,sizeof(file),1,fp)==1)
            {
                if(strcmp(file.time,time)!=0)
                    fwrite(&file,sizeof(file),1,fptr);
            }
            fclose(fp);
            fclose(fptr);
            remove(filename);
            rename("temp",filename);
            printf("\nDELETED SUCCESFULLY...");
            break;
        default:
```

```
                printf("\n\tYOU ENTERED WRONG CHOICE");

                break;

            }

        }
        while(choice<1||choice>2);

        printf("\n\tDO YOU LIKE TO DELETE ANOTHER RECORD.(Y/N):");

        fflush(stdin);

        scanf("%c",&another);

    }

    printf("\n\n\tPRESS ANY KEY TO EXIT...");

    getch();

}
```