

```

#include<fstream.h>
#include<conio.h>
#include<stdio.h>
#include<process.h>
#include<graphics.h>
#include<dos.h>
#include<string.h>

//STRUCTURE DATE
struct dat
{
    char dd[2],mm[2],yy[2];
}idate;
//END OF STRUCTURE

//FUNCTION TO GET DATE
void date()
{
    strcpy(idate.dd,"00");
    strcpy(idate.mm,"00");
    strcpy(idate.yy,"00");
    cout<<" dd: ";
    gets(idate.dd);
    cout<<"\t\t\tmm: ";
    gets(idate.mm);
    cout<<"\t\t\tty: ";
    gets(idate.yy);
}
//END OF FUNCTION

int midx;
int midy;

//CLASS BOOKS
class Books
{
    int idno;          //ID No. of book
    char name[35];     //Name of book
    char sec[20];      //section of library in which the book is kept
    char author[30];   //name of author
    int qty;           //quantity of books
    int bissue;        //no. of books issued
    int aqty;          //no of books available to issue
    char lang[10];     //language of book

```

```

    int shelfno;           //shelf no. in which book is kept
    void c_aqty()
    {
        aqty=qty-bissue;
    }

public: Books()
{
    idno=0;
    strcpy(name,"Nil");
    strcpy(sec,"Nil");
    strcpy(author,"Nil");
    qty=0;
    bissue=0;
    aqty=0;
    strcpy(lang,"Nil");
    shelfno=0;
}

void enter()              //to enter the details of books
{
    cout<<"\nEnter the ID No. of the book : ";
    cin>>idno;
    cout<<"Enter the name of the book : ";
    gets(name);
    cout<<"Enter the section of library where the book is kept : ";
    gets(sec);
    cout<<"Enter the name of the author/publication of book : ";
    gets(author);
    cout<<"Enter the language in which book is printed : ";
    gets(lang);
    cout<<"Enter the shelf no. of book : ";
    cin>>shelfno;
    cout<<"Enter the quantity of that book in library : ";
    cin>>qty;
    cout<<"Enter the no. of books already issued : ";
    cin>>bissue;
    c_aqty();
}

void display()            //to display the details of books
{
    cout<<"\nThe ID No. of book : ";

```

```

        cout<<idno;
        cout<<"\nThe name of book : ";
        puts(name);
        cout<<"The section of library where it is kept : ";
        puts(sec);
        cout<<"The author/publication of book : ";
        puts(author);
        cout<<"Language in which book is printed : ";
        puts(lang);
        cout<<"Shelf no. in which book is kept : ";
        cout<<shelfno;
        cout<<"\nQuantity of that book : ";
        cout<<qty;
        cout<<"\nNo. of copies already issued of this book : ";
        cout<<bissue;
        cout<<"\nNo. of copies available in library: "<<aqty;
    }

    int returnidno()          //to return ID No.
    {
        return idno;
    }

    char* returnname()       //to return book name
    {
        return name;
    }

    int rqty()               //to return no. of books in library
    {
        return qty;
    }

    int raqty()              //to return no. of books available
    {
        return aqty;
    }

    void add(int x)
    {
        qty+=x;
        c_aqty();
    }

```

```

void del(int x)
{
    qty-=x;
    c_aqty();
}

void ret_book()
{
    --bissue;
    c_aqty();
}

void issueb()
{
    ++bissue;
    c_aqty();
}

char* rauthor()
{
    return author;
}
}B;
//END OF CLASS BOOKS

//CLASS MEMBERS
class Members
{
    int rno;           //roll no. of student
    char name[25];      //name of student
    char bissue[35];    //book issued by student
    int ndays;          //no. of days for which student kept the book
    int fine;           //charging fine
    char phno[11];      //phone no.
    char addr[50];       //address of student
    char mclass[6];     //student's class
    char isdate[9];     //issue date
    char rdate[9];      //return date

    void cal_fine()     //to calculate fine A/T no of days
    {
        if(ndays<=7)
            fine=0;
    }
}

```

```

        else if(ndays>7&&ndays<=14)
            fine=10;
        else if(ndays>14&&ndays<=21)
            fine=20;
        else if(ndays>21)
            fine=50;
    }

    void date()          //to enter issue date
    {
        isdate[0]=idate.dd[0];
        isdate[1]=idate.dd[1];
        isdate[2]='/' ;
        isdate[3]=idate.mm[0];
        isdate[4]=idate.mm[1];
        isdate[5]='/';
        isdate[6]=idate.yy[0];
        isdate[7]=idate.yy[1];
    }

    void retdate(int nd)    //to calculate return date A/T no. of days
    {
        char nm[2];
        if(idate.mm[0]=='0')
        {
            if(idate.mm[1]=='4' || idate.mm[1]=='6' || idate.mm[1]=='9')
            {
                nm[0]='3';
                nm[1]='0';
            }
            else if(idate.mm[1]=='2')
            {
                nm[0]='2';
                nm[1]='8';
            }
            else
            if(idate.mm[1]=='1' || idate.mm[1]=='3' || idate.mm[1]=='5' || idate.mm[1]=='7' || idate.mm[1]=='8')
            {
                nm[0]='3';
                nm[1]='1';
            }
        }
        else

```

```

{
    if(idate.mm[1]=='0' || idate.mm[1]=='2')
    {
        nm[0]='3';
        nm[1]='1';
    }
    else
    {
        nm[0]='3';
        nm[1]='0';
    }
}
for(int i=0;i<nd;i++)
{
    if(idate.mm[0]=='1' && idate.mm[1]=='2' && idate.dd[0]==nm[0] && idate.dd[1]==nm[1])
    {
        idate.mm[0]='0';
        idate.mm[1]='1';
        idate.dd[0]='0';
        idate.dd[1]='1';
        idate.yy[1]++;
    }
    else if(idate.dd[0]==nm[0] && idate.dd[1]==nm[1])
    {
        idate.dd[0]='0';
        idate.dd[1]='1';
        if(idate.mm[1]=='9')
        {
            idate.mm[0]++;
            idate.mm[1]='0';
        }
        else
            idate.mm[1]++;
    }
    else if(idate.dd[1]=='9')
    {
        idate.dd[0]++;
        idate.dd[1]='0';
    }
    else
        idate.dd[1]++;
}
}

```

```

        rdate[0]=idate.dd[0];
        rdate[1]=idate.dd[1];
        rdate[2]='/' ;
        rdate[3]=idate.mm[0];
        rdate[4]=idate.mm[1];
        rdate[5]='/';
        rdate[6]=idate.yy[0];
        rdate[7]=idate.yy[1];
    }
public: Members()
    {
        rno=0;
        strcpy(name,"Null");
        strcpy(mclass,"Null");
        strcpy(phno,"Null");
        strcpy(addr,"Null");
        strcpy(bissue,"Null");
        strcpy(isdate,"Null");
        strcpy(rdate,"Null");
        ndays=0;
        fine=0;
    }

void get()          //to enter details of students
{
    cout<<"\nEnter the roll no. of student : ";
    cin>>rno;
    cout<<"Enter the name of the student : ";
    gets(name);
    cout<<"Enter the class of student : ";
    gets(mclass);
    cout<<"Enter the phone no. of student : ";
    gets(phno);
    cout<<"Enter address of student : ";
    gets(addr);
    cout<<"Enter the book recently issued by student : ";
    gets(bissue);
    cout<<"Enter the date of book issued: ";
    ::date();
    date();
    cout<<"Enter the no. of days for which the student kept the book: ";
    cin>>ndays;
    retdate(ndays);
}

```

```

        cal_fine();
    }

void show()          //to display details of students
{
    cout<<"\nRoll no. of student : ";
    cout<<rno;
    cout<<"\nName of student : ";
    puts(name);
    cout<<"Class of student : ";
    puts(mclass);
    cout<<"Phone no. of the student : ";
    puts(phno);
    cout<<"Address of the student : ";
    puts(addr);
    cout<<"Name of the book issued recently : ";
    puts(bissue);
    cout<<"Date of book issued: ";
    puts(isdate);
    cout<<"Return date : ";
    puts(rdate);
    cout<<"No. of days for which the student kept the book : ";
    cout<<ndays;
    if(fine!=0)
        cout<<"\nFine:"<<fine<<endl;
}

int srno()          //to return rno of student
{
    return rno;
}

char* sname()       //to return name of student
{
    return name;
}

char* sclass()
{
    return mclass;
}

char* bname()       //to return name of book

```



```

{
    return bissue;
}

void ass_book()
{
    strcpy(bissue,"NULL");
}

void changedate()
{
    isdate[0]=isdate[1]=isdate[3]=isdate[4]=isdate[6]=isdate[7]=00;
    ndays=0;
    retdate(ndays);
    cal_fine();
}

void issue(char c[])
{
    strcpy(bissue,c);
    cout<<"Date of issuing new book:   ";
    ::date();
    date();
    cout<<"\nEnter the no. of days student keep the book: ";
    cin>>ndays;
    retdate(ndays);
    cal_fine();
}

void details()
{
    cout<<"Name: ";
    puts(name);
    cout<<"Book: ";
    puts(bissue);
    cout<<"Date of issue: ";
    puts(isdate);
    cout<<"\nFine: "<<fine;
}

void apply()
{
    cout<<"Roll no.: ";
    cin>>rno;
}

```

```

        cout<<"Name: ";
        gets(name);
        cout<<"Class: ";
        gets(mclass);
        cout<<"Address: ";
        gets(addr);
        cout<<"Phone no.: ";
        gets(phno);
        strcpy(bissue,"NULL");
        strcpy(isdate,"NULL");
        strcpy(rdate,"NULL");
        ndays=0;
        fine=0;
    }
}M;
//END OF CLASS MEMBERS

//FUNCTION TO CREATE FILE TO STORE DATA OF BOOKS
void create_books()
{
    char ch;
    ofstream f("books.dat",ios::binary);
    do
    {
        cout<<"\nEnter details of books:-";
        B.enter();
        f.write((char*)&B,sizeof B);
        cout<<"\nDo you want to enter the details of more books???"<
        cin>>ch;
        clrscr();
    }while(ch=='y'||ch=='Y');
    f.close();
}

//FUNCTION TO CREATE TO STORE DATA OF MEMBERS
void create_mem()
{
    char c;
    ofstream f("members.dat",ios::binary);
    do
    {
        cout<<"\nEnter details of student:-";
        M.get();
    }
}

```

```

        f.write((char*)&M,sizeof M);
        cout<<"\nDo you want to enter details of more students???"<
        cin>>c;
        clrscr();
    }while(c=='y'||c=='Y');
    f.close();
}

```

//FUNCTION TO APPLY FOR MEMBERSHIP OF LIBRARY

```

void apply_mem()
{
    int found=0;
    ifstream f("members.dat",ios::binary);
    ofstream fout("temp.dat",ios::binary);
    Members me;
    cout<<"\nEnter your necessary details:-\n";
    me.apply();
    while(!f.eof())
    {
        f.read((char*)&M,sizeof M);
        if(!f)
            break;
        if(me.srno()<M.srno())
        {
            fout.write((char*)&me,sizeof me);
            fout.write((char*)&M,sizeof M);
            found=1;
            break;
        }
        else
            fout.write((char*)&M,sizeof M);
    }
    if(found==0)
        fout.write((char*)&me,sizeof me);
    else
        while(!f.eof())
        {
            f.read((char*)&M,sizeof M);
            if(!f)
                break;
            fout.write((char*)&M,sizeof M);
        }
    f.close();
}

```

```
fout.close();
remove("members.dat");
rename("temp.dat","members.dat");
}
```

//FUNCTION TO DISPLAY DATA OF BOOKS

```
void display_books()
{
    ifstream f("books.dat",ios::binary);
    while(!f.eof())
    {
        f.read((char*)&B,sizeof B);
        if(!f)
            break;
        clrscr();
        cout<<"\nDetails of books\n";
        B.display();
        getch();
    }
    f.close();
}
```

//FUNCTION TO DISPLAY DATA OF MEMBERS

```
void display_mem()
{
    ifstream f("members.dat",ios::binary);
    while(f.read((char*)&M,sizeof M))
    {
        if(!f)
            break;
        clrscr();
        cout<<"\nDetails of student\n";
        M.show();
        getch();
    }
    f.close();
}
```

//FUNCTION TO INSERT MORE RECORDS OF BOOKS

```
void insert_books()
{
    int found=0;
    ifstream f("books.dat",ios::binary);
```

```

ofstream fout("temp.dat",ios::binary);
Books bi;
cout<<"\nEnter details of books : ";
bi.enter();
while(!f.eof())
{
    f.read((char*)&B,sizeof B);
    if(!f)
        break;
    if(bi.returnidno()<B.returnidno())
    {
        fout.write((char*)&bi,sizeof bi);
        fout.write((char*)&B,sizeof B);
        found=1;
        break;
    }
    else
        fout.write((char*)&B,sizeof B);
}
if(found==0)
    fout.write((char*)&bi,sizeof bi);
else
    while(!f.eof())
    {
        f.read((char*)&B,sizeof B);
        if(!f)
            break;
        fout.write((char*)&B,sizeof B);
    }
f.close();
fout.close();
remove("books.dat");
rename("temp.dat","books.dat");
}

```

//FUNCTION TO DELETE ONLY ONE BOOK FROM A RECORD OF BOOKS

```

void change_books()
{
    char c;
    ifstream f("books.dat",ios::binary);
    ofstream fout("temp.dat",ios::binary);
    char n[20];
    int i=0,x;

```

```

cout<<"\n\nEnter the name of book whose quantity is to be changed: ";
gets(n);

while(!f.eof())
{
    f.read((char*)&B,sizeof B);
    if(!f)
        break;
    if(strcmp(n,B.returnname())!=0)
        fout.write((char*)&B,sizeof B);
    else
    {
        cout<<"\nYou want to:-\n(A)Add some books\n(B)Delete some books\n";
        cout<<"Enter your choice: ";
        cin>>c;
        switch(c)
        {
            case 'a':
            case 'A':cout<<"Enter the no. of books to be added: ";
                    cin>>x;
                    B.add(x);
                    cout<<x<<" books have been added in the library";
                    cout<<"\nThank you";
                    break;
            case 'b':
            case 'B':cout<<"Enter no. of books to be deleted: ";
                    cin>>x;
                    B.del(x);
                    cout<<x<<" books have been deleted from the library";
                    cout<<"\nThank you";
                    break;
            default: cout<<"SORRY!!!Wrong choice";
        }
        i=1;
        fout.write((char*)&B,sizeof B);
        break;
    }
}
while(f.read((char*)&B,sizeof B))
{
    fout.write((char*)&B,sizeof B);
}
if(i==0)

```

```

        cout<<"No book with this name is in library";
f.close();
fout.close();
remove("books.dat");
rename("temp.dat","books.dat");
}

```

//FUNCTION TO INSERT MORE RECORDS OF MEMBERS

```

void insert_mem()
{
int found=0;
ifstream f("members.dat",ios::binary);
ofstream fout("temp.dat",ios::binary);
Members me;
cout<<"\nEnter details of member :\n ";
me.get();
while(!f.eof())
{
    f.read((char*)&M,sizeof M);
    if(!f)
        break;
    if(me.srno()<M.srno())
    {
        fout.write((char*)&me,sizeof me);
        fout.write((char*)&M,sizeof M);
        found=1;
        break;
    }
    else
        fout.write((char*)&M,sizeof M);
}
if(found==0)
    fout.write((char*)&me,sizeof me);
else
    while(!f.eof())
    {
        f.read((char*)&M,sizeof M);
        if(!f)
            break;
        fout.write((char*)&M,sizeof M);
    }
f.close();
fout.close();
}

```

```

remove("members.dat");
rename("temp.dat","members.dat");
}

```

//FUNCTION TO DELETE RECORDS OF MEMBERS

```

void delete_mem(int n)
{
ifstream f("members.dat",ios::binary);
ofstream fout("temp.dat",ios::binary);
int i=0;
while(!f.eof())
{
    f.read((char*)&M,sizeof M);
    if(!f)
        break;
    if(M.srno()!=n)
        fout.write((char*)&M,sizeof M);
    else
        i=1;
}
if(i==0)
    cout<<"No member with this S.No is registered";
else
    cout<<"Member deleted successfully";
f.close();
fout.close();
remove("members.dat");
rename("temp.dat","members.dat");
}

```

//FUNCTION TO DELELTE THE WHOLE RECORD OF BOOK FROM LIBRARY

```

void bookdelete()
{
ifstream f("books.dat",ios::binary);
ofstream fout("temp.dat",ios::binary);
char n[20];
int i=0;
cout<<"\nEnter the name of book whose data is to be deleted : ";
gets(n);
while(!f.eof())
{
    f.read((char*)&B,sizeof B);
    if(!f)

```



```

        break;
    if(strcmp(B.returnname(),n)!=0)
        fout.write((char*)&B,sizeof B);
    else
        i=1;
}
if(i==0)
    cout<<"No book with this name is in library";
else
    cout<<"Book deleted successfully";
f.close();
fout.close();
remove("books.dat");
rename("temp.dat","books.dat");
}

```

//FUNCTION TO RETURN BOOK TO LIBRARY

```

void return_book()
{
    int i=0;
    ifstream f("members.dat",ios::binary);
    ifstream t("books.dat",ios::binary);
    ofstream fout("temp.dat",ios::binary);
    ofstream tout("save.dat",ios::binary);
    char n[20];
    cout<<"Enter the name of student: ";
    gets(n);
    while(f.read((char*)&M,sizeof M))
    {
        if(!f)
            break;
        if(strcmp(n,M.sname())==0)
        {
            i=1;
            cout<<"Name of book: "<<M.bname();
            while(t.read((char*)&B,sizeof B))
            {
                if(!t)
                    break;
                if(strcmp(M.bname(),B.returnname())==0)
                {
                    B.ret_book();
                    tout.write((char*)&B,sizeof B);
                }
            }
        }
    }
}

```

```

        }
        else
            tout.write((char*)&B,sizeof B);
    }
    M.ass_book();
    M.changedate();
    fout.write((char*)&M,sizeof M);
}
else
    fout.write((char*)&M,sizeof M);
}
if(i==0)
    cout<<"\nYou have entered wrong name of member";
else
    cout<<"\nYour book has been successfully returned.\nTHNAK YOU!!!!";
f.close();
t.close();
fout.close();
tout.close();
remove("members.dat");
remove("books.dat");
rename("temp.dat","members.dat");
rename("save.dat","books.dat");
}

```

//FUNCTION TO ISSUE BOOK FROM LIBRARY

```

void issue_book()
{
    int i=0,j=0;
    ifstream f("members.dat",ios::binary);
    ifstream t("books.dat",ios::binary);
    ofstream fout("temp.dat",ios::binary);
    ofstream tout("save.dat",ios::binary);
    char n[20],b[20],x;
    cout<<"\nEnter the name of the student: ";
    gets(n);
    while(f.read((char*)&M,sizeof M))
    {
        if(!f)
            break;
        if(strcmp(n,M.sname())==0)
            t:{
                cout<<"Enter the name of book the student want to issue: ";

```

```

        gets(b);
        while(t.read((char*)&B,sizeof B))
        {
            if(!t)
                break;
            if(strcmp(M.bname(),B.returnname())==0)
            {
                B.ret_book();
                tout.write((char*)&B,sizeof B);
            }
            if(strcmp(b,B.returnname())==0)
            {
                B.issueb();
                tout.write((char*)&B,sizeof B);
                j=1;
                break;
            }
            else
            {
                tout.write((char*)&B,sizeof B);
            }
        }
        M.issue(b);
        fout.write((char*)&M,sizeof M);
        i=1;
        break;
    }
    else
    {
        fout.write((char*)&M,sizeof M);
    }
}
if(i==0)
    cout<<"No member with this name is registered";
if(j==0)
{
    cout<<"No book with this name is in library ";
    cout<<"\n\nDo you want another book???"';
    cin>>x;
    if(x=='y' || x=='Y')
        goto t;
}
else

```

```

{
    cout<<"Your book has been issued successfully\n";
    cout<<"Now you have to return the book in the given time otherwise you have to pay for
it";
    M.ass_book();
}
f.close();
t.close();
fout.close();
tout.close();
remove("members.dat");
remove("books.dat");
rename("temp.dat","members.dat");
rename("save.dat","books.dat");
}

```

//FUNCTION TO SEARCH MEMBER BY NAME

```

void search_mem(char n[])
{
    int i=0;
    ifstream f("members.dat",ios::binary);
    while(f.read((char*)&M,sizeof M))
    {
        if(!f)
            break;
        if(strcmp(n,M.sname())==0)
        {
            M.show();
            i=1;
        }
    }
    if(i==0)
        cout<<"No member with this is registered.";
    f.close();
}

```

//FUNCTION TO SEARCH MEMBER BY ROLL NO.

```

void search_mem(int n)
{
    int j=0;
    ifstream f("members.dat",ios::binary);
    while(f.read((char*)&M,sizeof M))
    {

```

```

        if(!f)
            break;
        if(n==M.srno())
        {
            M.show();
            j=1;
        }
    }
    if(j==0)
        cout<<"No member with this S.No. is registered.";
    f.close();
}

```

//FUNCTION TO SEARCH MEMBERS BY THEIR CLASS

```

void search(char n[])
{
    int i=0;
    ifstream f("members.dat",ios::binary);
    while(f.read((char*)&M,sizeof M))
    {
        if(!f)
            break;
        if(strcmp(n,M.sclass())==0)
        {
            M.show();
            getch();
            clrscr();
            i=1;
        }
    }
    if(i==0)
        cout<<"No member of this class is member of library.";
    f.close();
}

```

//FUNCTION TO SEARCH BOOK BY NAME

```

void search_book()
{
    int i=0;
    char n[35];
    ifstream f("books.dat",ios::binary);
    cout<<"Enter the name of book whose details to be searched: ";
    gets(n);
}

```

```

while(f.read((char*)&B,sizeof B))
{
    if(!f)
        break;
    if(strcmp(n,B.returnname())==0)
    {
        B.display();
        i=1;
    }
}
if(i==0)
    cout<<"There is no book with this name is in library.";
f.close();
}

```

```

//FUNCTION TO SEARCH BOOK BY AUTHOR NAME
void search_author()
{
    int j=0;
    char n[35];
    ifstream f("books.dat",ios::binary);
    cout<<"Enter the author of book whose details to be searched: ";
    gets(n);
    while(f.read((char*)&B,sizeof B))
    {
        if(!f)
            break;
        if(strcmp(n,B.rauthor())==0)
        {
            B.display();
            j=1;
        }
    }
    if(j==0)
        cout<<"There is no book of this author in library";
    f.close();
}

```

```

//FUNCTION WHICH THE LIBRARY CAN ACCESS
void authority()
{
    char ch,c;
    int a;

```

```

char password[7];
clrscr();
f:cout<<"Enter the password: ";
  for(int i=0;i<6;i++)
  {
      password[i]=getch();
      cout<<"*";
  }
  if(strcmp(password,"lfpsss")==0)
  {
      do
      {
          clrscr();
          cout<<"\n  1.Create a new file to store the records of books";
          cout<<"\n  2.Create a new file to store the records of students";
          cout<<"\n  3.Display the records of all the books";
          cout<<"\n  4.Display the records of all students";
          cout<<"\n  5.Insert more records of books";
          cout<<"\n  6.Change quantity of book";
          cout<<"\n  7.Delete the whole record of book";
          cout<<"\n  8.Search any record of book";
          cout<<"\n  9.Insert more records of students";
          cout<<"\n 10.Delete any record of student";
          cout<<"\n 11.Search any record of student";
          cout<<"\n 12.Help";
          cout<<"\n 13.Go back to main menu";
          cout<<"\n 14.Exit";
          cout<<"\nWhat do you want to do???\\nEnter your choice: ";
          cin>>a;
          clrscr();
          switch(a)
          {
              case 1:create_books();
                  break;
              case 2:create_mem();
                  break;
              case 3:display_books();
                  break;
              case 4:display_mem();
                  break;
              case 5:insert_books();
                  break;
              case 6:change_books();

```

```

        break;
    case 7:bookdelete();
        break;
    case 8:cout<<"You can search by:(A)Book name\n\t\t (B)Author name";
        cout<<"\nEnter your choice:";
        cin>>ch;
        switch(ch)
        {
            case 'a':
            case 'A':search_book();
                break;
            case 'b':
            case 'B':search_author();
                break;
        }
        break;
    case 9:insert_mem();
        break;
    case 10:int n;
        cout<<"\nEnter the roll no. of student whose record is to be deleted: ";
        cin>>n;
        delete_mem(n);
        break;
    case 11:cout<<"\nYou can search records of students by:\n\t(A)Roll
no.\n\t(B)Student name\n\t(C)Class";
        cout<<"\nEnter your choice:";
        cin>>ch;
        switch(ch)
        {
            case 'a':
            case 'A': int y;
                cout<<"Enter the roll no. of student whose details to be
searched: ";
                cin>>y;
                search_mem(y);
                break;
            case 'b':
            case 'B': char x[35];
                cout<<"Enter the name of student whose details to be
searched: ";
                gets(x);
                search_mem(x);
                break;

```



```

        case 'c':
        case 'C': char z[6];
                    cout<<"Enter the class of students whose data to
searched: ";

                    gets(z);
                    search(z);
                    break;
            }
            break;
        case 12:cout<<"The menu contains following options:";
                cout<<"\nCREATE FILE OF BOOKS:You can change all the records of
books by entering all\n necessary details.";
                cout<<"\nCREATE FILE OF MEMBERS:You can change all the records of
members by entering\n their all necessary details.";
                cout<<"\nDISPLAY DETAILS OF BOOKS:You can check all the details of
all books present\n in library.";
                cout<<"\nDISPLAY DETAILS OF MEMBERS:You can check all the details
of all mebers\n registered in library.";
                cout<<"\nINSERT BOOKS:You can add data of new books in library by
giving their\n necessary details.";
                cout<<"\nCHANGE QUANTITY OF BOOK:You can change quantity of
books as well if some\n copies of book are torn out or are too old.";
                cout<<"\nDELETE BOOK:You can delete all details of any book in library
if they are no\n more copies of that book.";
                cout<<"\nSEARCH BOOK:You can check all the details of any book in 2
ways:\n1.By its name\t\t2.By its author's name";
                cout<<"\nINSERT MEMBER:You can add data of new members of
applied to library.";
                cout<<"\nDELETE MEMBERS:You can delete all the details of any
member due to any kind of\n indiscipline action in library by them.";
                cout<<"\nSEARCH MEMBER:You can check all the details of any meber
in 3 ways:\n1.By his/her s.no.\t 2.By his/her name\t 3.By his/her class";
                break;
        case 13:goto g;
        case 14: exit(0);
        default:cout<<"\nOOPS!!!!WRONG CHOICE!!!!";
    }
    cout<<"\nDo you want to do something else???  ";
    cin>>c;
    }while(c=='y'||c=='Y');
    }
    else
    {

```

```

        cout<<"\n\n\tWRONG PASSWORD\nWant to try again???";
        cin>>ch;
        if(ch=='y' || ch=='Y')
            goto f;
    }
    g:
}
//FUNCTION THE STUDENT CAN ACCESS
void students()
{
    int a;
    char c;
    do
    {
        e:
        clrscr();
        cout<<" 1.Apply for membership\n";
        cout<<" 2.Request to issue book from the library\n";
        cout<<" 3.Return book to the library\n";
        cout<<" 4.Check your own details\n";
        cout<<" 5.Search required books\n";
        cout<<" 6.Complaint for anything in the library\n";
        cout<<" 7.Help\n";
        cout<<" 8.Go back to main menu\n";
        cout<<" 9.Exit\n";
        cout<<"What do you want to do??? \nEnter your choice:";
        cin>>a;
        clrscr();
        switch(a)
        {
            case 1:apply_mem();
                    cout<<"You are now a member of library";
                    break;
            case 2:issue_book();
                    break;
            case 3:return_book();
                    getch();
                    break;
            case 4:char n[20];
                    cout<<"\nEnter your name: ";
                    gets(n);
                    search_mem(n);
                    break;

```

```

case 5:cout<<"\nYou may search book by:\n(A)Book name\n(B)Author name";
      cout<<"\nEnter your choice: ";
      cin>>c;
      switch(c)
      {
          case 'a':
          case 'A':search_book();
                  break;
          case 'b':
          case 'B':search_author();
      }
      break;
case 6:char complaint[150];
      cout<<"\nYou can write your complaint following:\n->";
      gets(complaint);
      cout<<"THANKYOU.\nYour complaint has been recorded";
      cout<<"\nWe will definitely try to solve this problem";
      break;
case 7:cout<<"\nThe menu contains the following options";
      cout<<"\n 1.APPLY FOR MEMBERSHIP: You can give required details to the
library if\n want to become a member in library.";
      cout<<"\n 2.REQUEST FOR BOOK ISSUE: You can reuest for your wanted
book. If library\n has the book it will be issued to you, otherwise your current book will\n be
returned.";
      cout<<"\n 3.RETURN BOOK: You can return the book to the library when
your time is over\n with your clearing your fine.";
      cout<<"\n 4.CHECK YOUR DETAILS: You can check your details entered in
your library \naccount.";
      cout<<"\n 5.SEARCH BOOK: You can search the book you want.";
      cout<<"\n 6.COMPLAINT: You can complaint about any other student who is
destroying\n the books in library or creating indiscipline in the library or you can also\n give your
opinion for any book you want in the library";
      break;
case 8:goto g;
case 9:exit(0);
default:cout<<"\nOOPS!!!!WRONG CHOICE!!!!";
      }
      cout<<"\nDo you want to do something else???"
      cin>>c;
}while(c=='y'||c=='Y');
      g:
}

```

```
//FUNCTION TO DISPLAY WELCOME PAGE
```

```
void welcome()
```

```
{
    int gdriver = DETECT, gmode, errorcode;
    initgraph(&gdriver, &gmode, "c:\\turbo3\\bgi");
    midx=getmaxx()/2;
    midy=getmaxy()/2;
    setcolor(GREEN);
    settxtstyle(BOLD_FONT,HORIZ_DIR,3);
    outtextxy(midx-3,midy-2,"WELCOME:-");
    errorcode = graphresult();
    if (errorcode != grOk)
    {
        cout<<"Graphics error: %s\n";
        cout<<grapherrormsg(errorcode);
        cout<<"Press any key to halt:";
        getch();
        exit(1);
    }
    getch();
    closegraph();
}
```

```
void main()
```

```
{
    clrscr();
    int n;
    char c;
    welcome();
    delay(300);
    clrscr();
    textbackground(CYAN);
    textcolor(WHITE);
    do
    {
        clrscr();
        cout<<"\\t\\t\\tMAIN MENU";
        cout<<"\\n 1.Teacher\\n 2.Student\\n 3.Help\\n 4.Exit\\n";
        cin>>n;
        if(n==1)
        {
            textbackground(GREEN);
            textcolor(WHITE);
```

```

        authority();
    }
    else if(n==2)
    {
        textbackground(BLUE);
        textcolor(WHITE);
        students();
    }
    else if(n==3)
    {
        clrscr();
        textbackground(YELLOW);
        textcolor(BLACK);
        gotoxy(1,10);
        cout<<"This is program is to manage a library of school.";
        gotoxy(1,11);
        cout<<"Here are 2 ways to access this program.";
        gotoxy(1,12);
        cout<<"1.Authority";
        gotoxy(1,13);
        cout<<"2.Students";
        gotoxy(1,14);
        cout<<"And more further information is given ";
        cout<<"within the functions.";
        gotoxy(10,16);
        cout<<"THANK YOU";
        getch();
    }
    else if(n==4)
    {
        exit(0);
    }
    else
        cout<<"\nSORRY!!!!WRONG CHOICE!!!!";

    clrscr();
    cout<<"\nDo you want to go back to Main Menu???";
    cin>>c;
    textbackground(CYAN);
    }while(c=='y'||c=='Y');
    exit(0);
    getch();
}

```