DR. B.R. AMBEDKAR NATIONAL INSTITUTE OF TECHNOLOGY, JALANDHAR (144011), PUNJAB

Database Management System Lab

CSPC-222

Lab 1

Submitted by: Bhart Bansal
Roll. No.: 20103039
Department: CSE
Group: A2
Sem: 4th
Submitted to: Dr. Rajneesh Rani
Date: 18/01/2022

Database Management Systems Lab

CSPC-222

Lab-Assignment 1

Using C++, WAP of a Library Management Systems with the concepts of File Handling. It must Allow the Library to Manage the records of Issues and Deposits of Books and Add, Retrieve, and update Details of any Student or Book.

The basic features the library management system must include are:

- **Admin Access**
  Create Separate Records/files for Students and Books
  - In Students File, store: Student Name, Admission Number, Class, No of Books Issued.
  - In Books File, store: Book Number, Author name, Book Title, Domain, Edition and No of Copies Available.

  The admin has access to:

  - Add new Student and Book records.
  - Modify any Student or Book record.
  - Delete any Student or Book record.
  - View any Specific or Whole Student record and Book record.
- **Book Issue**: Used at time of Issuing A book. Takes in input as Book Number and Issuer's Admission number, issue date. Adds that detail to the student record that the specific Student has issued a book. Create a file which stores this record of Book Issue.
- **Book Deposit**: When Student returns the book, it uses student Admission Number and Book Number to modify the record of Book issued by the student. Create a file which stores this record of Book Deposit.

  Note: A book Cannot be issued if there are no more copies available and a student can only issue 1 copy of a book and at max 3 books at a time.
  A book can only be issued for 15 days at once**.** After that it needs to be re-issued. If the book is deposited after 15 days of issue, a fine is imposed on per day basis.

```
#include <bits/stdc++.h>
using namespace std;
#define endl "\n"
#define fine 10

const int monthDays[12] = {31, 59, 90, 120, 151, 181, 212, 243, 273, 304,
334, 365};
int countLeapYearDays(int d[])
{
    int years = d[2];
    if (d[1] <= 2)
        years--;
    return ((years / 4) - (years / 100) + (years / 400));
}
int countNoOfDays(int d1, int d2)
{
    int date1[3], date2[3];
    date1[0] = d1 / 1000000;
    d1 = d1 % 1000000;
    date1[1] = d1 / 10000;
    d1 = d1 % 10000;
    date1[2] = d1;
    date2[0] = d2 / 1000000;
    d2 = d2 % 1000000;
    date2[1] = d2 / 10000;
    d2 = d2 % 10000;
    date2[2] = d2;
    long long int dayCount1 = (date1[2] * 365);
    dayCount1 += monthDays[date1[1] - 1];
    dayCount1 += date1[0];
    dayCount1 += countLeapYearDays(date1);
    long long int dayCount2 = (date2[2] * 365);
    dayCount2 += monthDays[date2[1] - 1];
    dayCount2 += date2[0];
    dayCount2 += countLeapYearDays(date2);
    return (abs(dayCount1 - dayCount2));
}


class Book
{
public:
    int Book_Number;
    char Author_name[20], Book_Title[20], Domain[20];
    int Edition, Copies_Available;


    void get_details()
    {
        cout << "Enter Book Number: ";
        cin >> Book_Number;
        cin.ignore();
        cout << "Enter Author Name: ";
        cin.getline(Author_name, 20);
        cout << "Enter Book Title: ";
        cin.getline(Book_Title, 20);
        cout << "Enter Domain: ";
        cin.getline(Domain, 20);
        cout << "Enter Edition: ";
        cin >> Edition;
        cout << "Enter Copies Available: ";
        cin >> Copies_Available;
        cout << endl;
    }

    void show_details()
    {
        cout << "Book Number: " << Book_Number << endl;
        cout << "Author Name: " << Author_name << endl;
        cout << "Book Title: " << Book_Title << endl;
        cout << "Domain: " << Domain << endl;
        cout << "Edition: " << Edition << endl;
        cout << "Copies Available: " << Copies_Available << endl;
    }
};

class Issue
{

public:
    int Book_Number, Admission_number, issue_date;
    Issue()
    {
        Book_Number = 0;
        Admission_number = 0;
        issue_date = 0;
    }
    void get_details()
    {
        cout << "Enter Book Number: ";
        cin.ignore();
        cin >> Book_Number;
        cout << "Enter Admission Number: ";
        cin >> Admission_number;
        cout << "Enter date((as int)DDMMYYYY): ";
        cin >> issue_date;
    }
    void show_details()
    {
        cout << "Book Number: " << Book_Number << endl;
        cout << "Admission Number: " << Admission_number << endl;
        cout << "Date((as int)DDMMYYYY): " << issue_date << endl;
    }
};

class Student
{
public:
    char Student_Name[20];
    int Admission_Number, Class, Books_Issued;
```

```
Issue book[3];


void get_details()
{
    cout << "Enter Admission Number: ";
    cin >> Admission_Number;
    cout << "Enter Student Name: ";
    cin.ignore();
    cin.getline(Student_Name, 20);
    cout << "Enter Class: ";
    cin >> Class;
    cout << "Enter Books Issued: ";
    cin >> Books_Issued;
    cout << "\n";
    for (int i = 0; i < Books_Issued; i++)
    {
        book[i].get_details();
        cout << "\n";
    }
}
void show_details()
{
    cout << "Student Name: " << Student_Name << endl;
    cout << "Admission Number: " << Admission_Number << endl;
    cout << "Class: " << Class << endl;
    cout << "Books Issued: " << Books_Issued << endl;
    booksIssuedDetails();
    }
    void booksIssuedDetails()
    {
        if (Books_Issued == 0)
            cout << "No book issued\n";
        else
        {
            cout << "\nTotal books issued: " << Books_Issued << endl;
            for (int i = 0; i < Books_Issued; i++)
            {
                book[i].show_details();
                cout << endl;
            }
        }
        cout << "**************************************\n";
    }
};


class library
{
    Book searchBook(fstream &book, int num)
    {
        Book b;
        while (book.read((char *)&b, sizeof(Book)))
        {
            if (num == b.Book_Number)
            {
```

```
            return b;
        }
    }
    return b;
}
Student searchStudent(fstream &student, int num)
{
    Student s;
    while (student.read((char *)&s, sizeof(Student)))
    {
        if (num == s.Admission_Number)
        {
            return s;
        }
    }
    return s;
}
void addBook(fstream &book)
{
    Book b;
    b.get_details();
    book.write((char *)&b, sizeof(Book));
}
void addStudent(fstream &student)
{
    Student s;
    s.get_details();
    student.write((char *)&s, sizeof(Student));
```

```
}



void issueBook(int adNum, int bNum)
{
    Book b;
    ifstream book("Books.txt", ios::binary);
    fstream t;
    t.open("temp.txt", ios::binary | ios::out | ios::app);
    while (book.read((char *)&b, sizeof(Book)))
    {
        if (b.Book_Number == bNum)
        {
            b.Copies_Available--;
        }
        t.write((char *)&b, sizeof(Book));
    }
    book.close();
    t.close();
    remove("Books.txt");
    rename("temp.txt", "Books.txt");


    Student s;
    ifstream student("Students.txt", ios::in | ios::binary);
    fstream t1;
    t1.open("temp1.txt", ios::binary | ios::out | ios::app);
    while (student.read((char *)&s, sizeof(Student)))
    {
        if (s.Admission_Number == adNum)
        {
            s.book[s.Books_Issued].Admission_number = adNum;
            s.book[s.Books_Issued].Book_Number = bNum;
            cout << "Enter date((as int)DDMMYYYY): ";
            int x;
            cin >> x;
            s.book[s.Books_Issued].issue_date = x;
            s.Books_Issued++;;
        }
        t1.write((char *)&s, sizeof(Student));
    }
    student.close();
    t1.close();
    remove("Students.txt");
    rename("temp1.txt", "Students.txt");
}


void depositBook(int adNum, int bNum)
{
    Book b;
    ifstream book("Books.txt", ios::binary);
    fstream t;
    t.open("temp.txt", ios::binary | ios::out | ios::app);
    while (book.read((char *)&b, sizeof(Book)))
    {
        if (b.Book_Number == bNum)
        {
            b.Copies_Available++;
        }
        t.write((char *)&b, sizeof(Book));
    }
    book.close();
    t.close();
    remove("Books.txt");
    rename("temp.txt", "Books.txt");


    Student s;
    ifstream student("Students.txt", ios::in | ios::binary);
    fstream t1;
    t1.open("temp1.txt", ios::binary | ios::out | ios::app);
    int issued_day;
    while (student.read((char *)&s, sizeof(Student)))
    {
        if (s.Admission_Number == adNum)
        {
            bool flag = flag;
            int pos;
            for (int i = 0; i < s.Books_Issued; i++)
            {
                if (bNum == s.book[i].Book_Number)
                {
                    pos = i;
                    issued_day = s.book[i].issue_date;
```

```
                        flag = true;
                        break;
                    }
                }
                if (flag)
                {
                    s.book[pos].Admission_number = adNum;
                    s.book[pos].Book_Number = s.book[s.Books_Issued -
1].Book_Number;
                    s.book[pos].issue_date = s.book[s.Books_Issued -
1].issue_date;
                    s.Books_Issued--;
                }
                else
                    cout << "You do not have the book\n";
            }
            t1.write((char *)&s, sizeof(Student));
        }
        student.close();
        t1.close();
        remove("Students.txt");
        rename("temp1.txt", "Students.txt");


        cout << "Enter today dates: ";
        int day;
        cin >> day;
        int days = countNoOfDays(day, issued_day);
        if (days > 15)
        {
            cout << "Late deposit fine: " << (days - 15) * fine << endl;
        }
        else
        {
            cout << "Book successfully deposited!!" << endl;
        }
    }


public:
    void greet()
    {
        cout << "\n****Welcome to the Library****\n\n";
        cout << "Please choose an option:\n";
        cout << "1 TO ADD/Return " << endl;
        cout << "2 TO SHOW" << endl;
        cout << "3 TO SEARCH" << endl;
        cout << "4 TO MODIFY" << endl;
        cout << "5 TO DELETE" << endl;
        cout << "6 TO Issue" << endl;
        cout << "7 TO Deposit" << endl;
        cout << "8 TO EXIT" << endl;
    }
    void add();
    void show();
    bool search();
    bool modify();
    bool del_record();
    void issue();
    void deposit();
};


void library ::add()
{
    int x;
    cout << "\nEnter 1 to add book, 2 to add student: ";
    cin >> x;
    if (x == 1)
    {
        fstream book;
        book.open("Books.txt", ios::binary | ios::out | ios::app);
        addBook(book);
        book.close();
    }


    if (x == 2)
    {
        fstream student;
        student.open("Students.txt", ios::binary | ios::out | ios::app);
        addStudent(student);
        student.close();
    }
}
```

```
void library ::show()
{
    int x;
    cout << "\nEnter 1 to show books, 2 to show students: ";
    cin >> x;
    if (x == 1)
    {
        Book b;
        ifstream book("Books.txt", ios::in | ios::binary);
        while (book.read((char *)&b, sizeof(Book)))
        {
            b.show_details();
            cout << endl;
        }
        book.close();
    }

    if (x == 2)
    {
        Student s;
        ifstream student("Students.txt", ios::in | ios::binary);
        while (student.read((char *)&s, sizeof(Student)))
        {
            s.show_details();
            cout << endl;
        }
        student.close();
    }
}


bool library ::search()
{
    int x;
    cout << "\nEnter 1 to search books list, 2 to search students list: ";
    cin >> x;
    bool flag = false;
    if (x == 1)
    {
        int num;
        cout << "Enter book number of book to search: ";
        cin >> num;
        Book b;
        fstream book;
        book.open("Books.txt", ios::in | ios::binary);
        b = searchBook(book, num);
        if (b.Book_Number == num)
        {
            b.show_details();
            flag = true;
        }
        cout << "\n";
        book.close();
    }
    if (x == 2)
    {
        int num;
        cout << "Enter admission number of student to search: ";
        cin >> num;
        Student s;
        fstream student;
        student.open("Students.txt", ios::in | ios::binary);
        s = searchStudent(student, num);
        if (s.Admission_Number == num)
        {
            s.show_details();
            flag = true;
        }
        cout << "\n";
        student.close();
    }
    return flag;
}


bool Library ::modify()
{
    int x;
    cout << "\nEnter 1 to modify books list, 2 to modify students list: ";
    cin >> x;
    bool flag = false;
```

```
if (x == 1)
{
    int bNum;
    cout << "Enter book number of book to modify: ";
    cin >> bNum;
    Book b;
    ifstream book("Books.txt", ios::binary);
    fstream t;
    t.open("temp.txt", ios::binary | ios::out | ios::app);
    while (book.read((char *)&b, sizeof(Book)))
    {
        if (bNum == b.Book_Number)
        {
            addBook(t);
            flag = true;
        }
        else
        {
            t.write((char *)&b, sizeof(Book));
        }
    }
    book.close();
    t.close();
    remove("Books.txt");
    rename("temp.txt", "Books.txt");
}
```

```
if (x == 2)
{
    int adNum;
    cout << "Enter Admission number of student to modify: ";
    cin >> adNum;
    Student s;
    ifstream student("Students.txt", ios::in | ios::binary);
    fstream t;
    t.open("temp.txt", ios::binary | ios::out | ios::app);
    while (student.read((char *)&s, sizeof(Student)))
    {
        if (adNum == s.Admission_Number)
        {
            addStudent(t);
            flag = true;
        }
        else
        {
            t.write((char *)&s, sizeof(Student));
        }
    }
    cout << "\n";
    student.close();
    t.close();
    remove("Students.txt");
    rename("temp.txt", "Students.txt");
}
return flag;
```

```
}


bool Library ::del_record()
{
    int x;
    cout << "\nEnter 1 to delete book record, 2 to delete student record: ";
    cin >> x;
    bool flag = false;
    if (x == 1)
    {
        int bNum;
        cout << "Enter book number of book to delete: ";
        cin >> bNum;
        Book b;
        ifstream book("Books.txt", ios::binary);
        fstream t;
        t.open("temp.txt", ios::binary | ios::out | ios::app);
        while (book.read((char *)&b, sizeof(Book)))
        {
            if (bNum == b.Book_Number)
            {
                flag = true;
            }
            else
            {
                t.write((char *)&b, sizeof(Book));
            }
        }
        book.close();
        t.close();
        remove("Books.txt");
        rename("temp.txt", "Books.txt");
    }


    if (x == 2)
    {
        int adNum;
        cout << "Enter Admission number of student to delete: ";
        cin >> adNum;
        Student s;
        ifstream student("Students.txt", ios::in | ios::binary);
        fstream t;
        t.open("temp.txt", ios::binary | ios::out | ios::app);
        while (student.read((char *)&s, sizeof(Student)))
        {
            if (s.Admission_Number == adNum)
            {
                flag = true;
            }
            else
            {
                t.write((char *)&s, sizeof(Student));
            }
        }
```

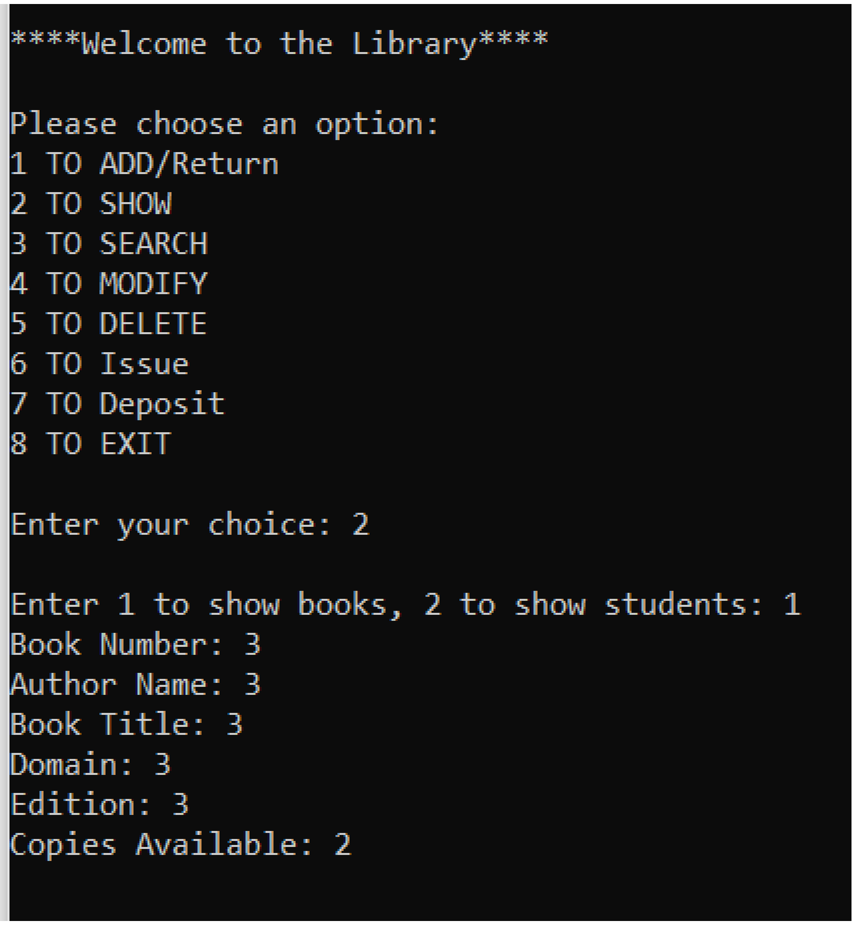
```
        student.close();
        t.close();
        remove("Students.txt");
        rename("temp.txt", "Students.txt");
    }
    return flag;
}


void Library::issue()
{
    int bNum, adNum;
    cout << "Enter your Admission number: ";
    cin >> adNum;
    cout << "Enter Book number to issue: ";
    cin >> bNum;
    fstream student;
    student.open("Students.txt", ios::binary | ios::in);
    Student s = searchStudent(student, adNum);
    student.close();
    fstream book;
    book.open("Books.txt", ios::binary | ios::in);
    Book b = searchBook(book, bNum);
    book.close();
    if (s.Admission_Number == adNum)
    {
        bool flag = false;
        if (s.Books_Issued == 3)
```

```
            cout << "You already have 3 books. You cannot issue more" <<
endl;
        else
        {
            flag = true;
            for (auto i : s.book)
            {
                if (i.Book_Number == bNum)
                {
                    cout << "You already have that book!!\n";
                    flag = false;
                    break;
                }
            }
            if (flag)
                if (b.Book_Number == bNum && b.Copies_Available > 0)
                    flag = true;
                else
                {
                    cout << "Book not available\n";
                    flag = false;
                }
        }
        if (flag)
        {
            issueBook(adNum, bNum);
        }
    }
    else
        cout << "No student record found!!\n";
}


void Library::deposit()
{
    int bNum, adNum;
    cout << "Enter your Admission number: ";
    cin >> adNum;
    cout << "Enter Book number to deposit: ";
    cin >> bNum;
    fstream student;
    student.open("Students.txt", ios::binary | ios::in);
    Student s = searchStudent(student, adNum);
    student.close();
    fstream book;
    book.open("Books.txt", ios::binary | ios::in);
    Book b = searchBook(book, bNum);
    book.close();
    if (s.Admission_Number == adNum)
    {
        bool flag = false;
        for (auto i : s.book)
        {
            if (i.Book_Number == bNum)
            {
                flag = true;
                break;
            }
        }
        if (b.Book_Number == bNum && flag)
        {
            depositBook(adNum, bNum);
        }
        else
            cout << "Wrong Book Number\n";
    }
    else
        cout << "No student record found!!\n";
}


int main()
{
    Library lib;
    int choice;
    do
    {
        lib.greet();
        cout << "\nEnter your choice: ";
        cin >> choice;


        switch (choice)
        {
        case 1:
```

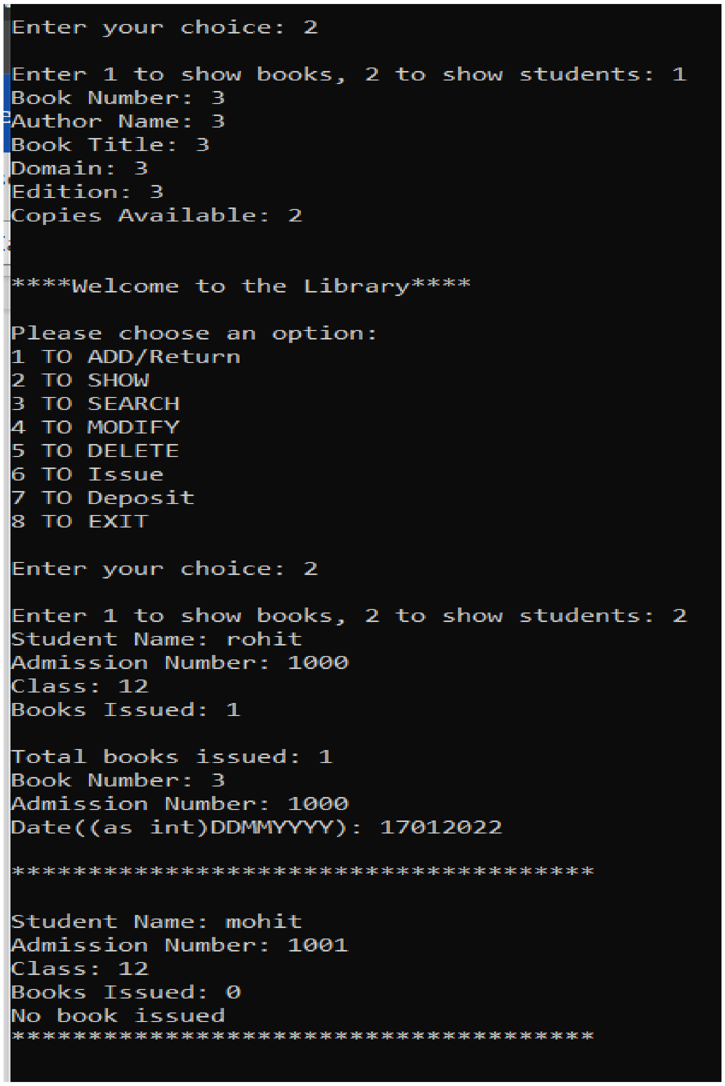
```
        lib.add();

        break;

    case 2:

        lib.show();

        break;

    case 3:

        if (!lib.search())

            cout << "Not Found\n";

        break;

    case 4:

        if (!lib.modify())

            cout << "Not Found\n";

        break;

    case 5:

        if (lib.del_record())

            cout << "Record Deleted\n";

        else

            cout << "Not Found\n";

        break;

    case 6:

        lib.issue();

        break;

    case 7:

        lib.deposit();

        break;

    }

} while (choice < 8);

cout << "\n**Thanks for the visit**\n\n";
```

```
    return 0;
}
```

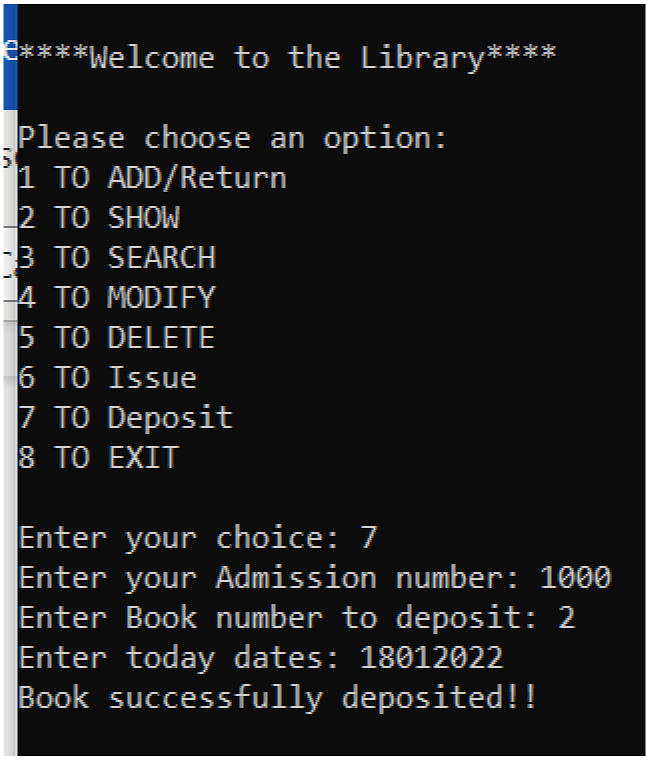
Output:



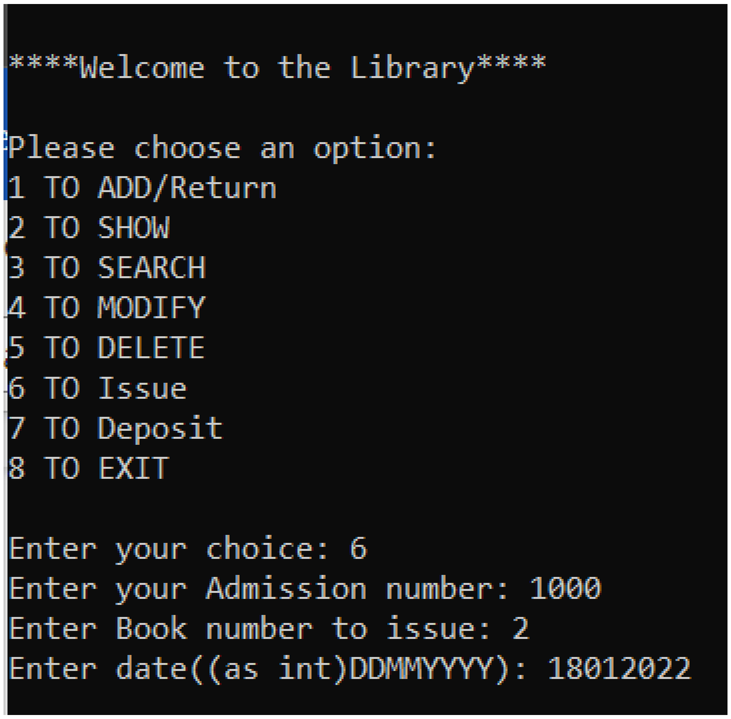
```
****Welcome to the Library****

Please choose an option:
1 TO ADD/Return
2 TO SHOW
3 TO SEARCH
4 TO MODIFY
5 TO DELETE
6 TO Issue
7 TO Deposit
8 TO EXIT

Enter your choice: 2

Enter 1 to show books, 2 to show students: 1
Book Number: 3
Author Name: 3
Book Title: 3
Domain: 3
Edition: 3
Copies Available: 2
```



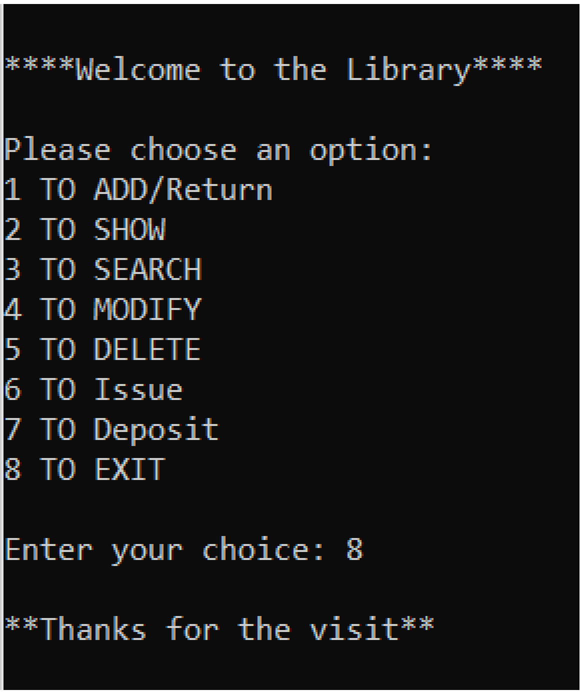
```
Enter your choice: 2

Enter 1 to show books, 2 to show students: 1
Book Number: 3
Author Name: 3
Book Title: 3
Domain: 3
Edition: 3
Copies Available: 2


****Welcome to the Library****

Please choose an option:
1 TO ADD/Return
2 TO SHOW
3 TO SEARCH
4 TO MODIFY
5 TO DELETE
6 TO Issue
7 TO Deposit
8 TO EXIT

Enter your choice: 2

Enter 1 to show books, 2 to show students: 2
Student Name: rohit
Admission Number: 1000
Class: 12
Books Issued: 1

Total books issued: 1
Book Number: 3
Admission Number: 1000
Date((as int)DDMMYYYY): 17012022

**************************************

Student Name: mohit
Admission Number: 1001
Class: 12
Books Issued: 0
No book issued
**************************************
```



```
****Welcome to the Library****

Please choose an option:
1 TO ADD/Return
2 TO SHOW
3 TO SEARCH
4 TO MODIFY
5 TO DELETE
6 TO Issue
7 TO Deposit
8 TO EXIT

Enter your choice: 7
Enter your Admission number: 1000
Enter Book number to deposit: 2
Enter today dates: 18012022
Book successfully deposited!!
```



```
****Welcome to the Library****

Please choose an option:
1 TO ADD/Return
2 TO SHOW
3 TO SEARCH
4 TO MODIFY
5 TO DELETE
6 TO Issue
7 TO Deposit
8 TO EXIT

Enter your choice: 6
Enter your Admission number: 1000
Enter Book number to issue: 2
Enter date((as int)DDMMYYYY): 18012022
```



```
****Welcome to the Library****

Please choose an option:
1 TO ADD/Return
2 TO SHOW
3 TO SEARCH
4 TO MODIFY
5 TO DELETE
6 TO Issue
7 TO Deposit
8 TO EXIT

Enter your choice: 8

**Thanks for the visit**
```