#include <iostream>

#include <fstream>

#include <vector>

#include <string>

using namespace std;

class Book {

private:

int bookID;

string title;

string author;

bool isAvailable;

public:

Book() {}

Book(int id, string t, string a, bool avail = true)

: bookID(id), title(t), author(a), isAvailable(avail) {}

int getID() const { return bookID; }

string getTitle() const { return title; }

string getAuthor() const { return author; }

bool getAvailability() const { return isAvailable; }

void issueBook() {

if (isAvailable) {

isAvailable = false;

cout << "Book issued successfully.\n";

} else {

cout << "Book is already issued.\n";

}

}

void returnBook() {

if (!isAvailable) {

isAvailable = true;

cout << "Book returned successfully.\n";

} else {

cout << "Book was not issued.\n";

}

}

string toString() const {

return to\_string(bookID) + "|" + title + "|" + author + "|" + (isAvailable ? "1" : "0");

}

static Book fromString(const string& data) {

int id;

string t, a;

int avail;

size\_t pos1 = data.find("|");

size\_t pos2 = data.find("|", pos1 + 1);

size\_t pos3 = data.find("|", pos2 + 1);

id = stoi(data.substr(0, pos1));

t = data.substr(pos1 + 1, pos2 - pos1 - 1);

a = data.substr(pos2 + 1, pos3 - pos2 - 1);

avail = stoi(data.substr(pos3 + 1));

return Book(id, t, a, avail == 1);

}

void display() const {

cout << "ID: " << bookID

<< " | Title: " << title

<< " | Author: " << author

<< " | Status: " << (isAvailable ? "Available" : "Issued") << endl;

}

};

class Library {

private:

vector<Book> books;

void loadFromFile() {

ifstream file("books.txt");

string line;

while (getline(file, line)) {

if (!line.empty()) {

books.push\_back(Book::fromString(line));

}

}

file.close();

}

void saveToFile() {

ofstream file("books.txt", ios::trunc);

for (const auto& b : books) {

file << b.toString() << endl;

}

file.close();

}

public:

Library() {

loadFromFile();

}

~Library() {

saveToFile();

}

void addBook() {

int id;

string title, author;

cout << "Enter Book ID: ";

cin >> id;

cin.ignore();

cout << "Enter Title: ";

getline(cin, title);

cout << "Enter Author: ";

getline(cin, author);

books.push\_back(Book(id, title, author));

cout << "Book added successfully.\n";

}

void issueBook() {

int id;

cout << "Enter Book ID to issue: ";

cin >> id;

for (auto& b : books) {

if (b.getID() == id) {

b.issueBook();

return;

}

}

cout << "Book not found.\n";

}

void returnBook() {

int id;

cout << "Enter Book ID to return: ";

cin >> id;

for (auto& b : books) {

if (b.getID() == id) {

b.returnBook();

return;

}

}

cout << "Book not found.\n";

}

void searchBook() {

int choice;

cout << "Search by 1. ID 2. Title : ";

cin >> choice;

cin.ignore();

if (choice == 1) {

int id;

cout << "Enter ID: ";

cin >> id;

for (auto& b : books) {

if (b.getID() == id) {

b.display();

return;

}

}

} else {

string title;

cout << "Enter Title: ";

getline(cin, title);

for (auto& b : books) {

if (b.getTitle() == title) {

b.display();

return;

}

}

}

cout << "Book not found.\n";

}

void showStatistics() {

int total = books.size();

int issued = 0, available = 0;

for (auto& b : books) {

if (b.getAvailability())

available++;

else

issued++;

}

cout << "Total Books: " << total

<< " | Available: " << available

<< " | Issued: " << issued << endl;

}

void displayAll() {

for (auto& b : books) {

b.display();

}

}

};

int main() {

Library lib;

int choice;

do {

cout << "\n--- Library Management System ---\n";

cout << "1. Add Book\n";

cout << "2. Issue Book\n";

cout << "3. Return Book\n";

cout << "4. Search Book\n";

cout << "5. Show Statistics\n";

cout << "6. Display All Books\n";

cout << "0. Exit\n";

cout << "Enter your choice: ";

cin >> choice;

switch (choice) {

case 1: lib.addBook(); break;

case 2: lib.issueBook(); break;

case 3: lib.returnBook(); break;

case 4: lib.searchBook(); break;

case 5: lib.showStatistics(); break;

case 6: lib.displayAll(); break;

case 0: cout << "Exiting...\n"; break;

default: cout << "Invalid choice!\n";

}

} while (choice != 0);

return 0;

}