#include <iostream>

#include <iomanip>

#include <fstream>

#include <cstring>

using namespace std;

// struct of book details

typedef struct

{

char title[50];

char author[50];

char publisher[50];

char ISBN[50];

char price[50];

char edition[50];

char numOfEdition[50];

char numOfCopies[50];

char location[50];

char status[50];

}BOOK;

// prototype function

int getbooks(BOOK book[]);

void list(BOOK book[], int size);

void search(BOOK book[], int size);

void deleteRecord(BOOK book[], int size);

// main function

int main(void)

{

BOOK book[100]; // array of struct (each element contain all those variable in struct like title, author and so on)

int size;

int choice;

size = getbooks(book); // return the number of books, store in variable "size"

cout << "Welcome to the library book record system!" << endl;

do

{

cout << "Please choose an action from the following: \n1 - List\n2 - Search\n3 - Delete\n4 - Exit" << endl;

cin >> choice; // selection of action

cout << '\n';

switch (choice) {

case 1:

list(book, size); // call list function

cout << '\n';

break;

case 2:

search(book, size); // call search function

cout << '\n';

break;

case 3:

deleteRecord(book, size); // call deleteRecord function

size = getbooks(book); // read the file again after deletion of books

cout << '\n';

break;

case 4:

cout << "Thank you for using the library book record system!" << endl;

break;

default:

cout << "The option selected is not valid." << endl;

}

} while (choice != 4); // loop until user decide to exit

return 0;

}

// getbooks function

int getbooks(BOOK book[])

{

int size = 0;

int num = 0;

ifstream booksListFile("booksList.txt"); // open file for read

if (!booksListFile) { // test if file opens successfully

cout << "Error opening file.\n";

exit(100);

}

else

{

while (booksListFile) { // loop until end of file

booksListFile.getline(book[num].title, 50); // get information from txt file to put into array of structure

booksListFile.getline(book[num].author, 50);

booksListFile.getline(book[num].publisher, 50);

booksListFile.getline(book[num].edition, 50);

booksListFile.getline(book[num].price, 50);

booksListFile.getline(book[num].ISBN, 50);

booksListFile.getline(book[num].numOfEdition, 50);

booksListFile.getline(book[num].numOfCopies, 50);

booksListFile.getline(book[num].location, 50);

booksListFile.getline(book[num].status, 50);

booksListFile.ignore();

num++;

size++; // count the number of books to return to main function

}

}

booksListFile.close(); // close file

return size;

}

// list function

void list(BOOK book[], int size)

{

int num = 0;

int bookNum = 0;

cout << "Books List: \n" << endl;

cout << "No.\tTitle\t\t\t Author\t\t Publisher/Imprint\tEdition Price ISBN\t\t No.of Edition No.of Copies Location Status\n";

for (int num = 0; num < size; num++) { // loop through the array of struct to printout all books details

bookNum++; // count the number of book and display it on screen

cout << bookNum << '\t' << left << setw(25) << book[num].title << setw(17) << book[num].author;

cout << setw(22) << book[num].publisher << setw(10) << book[num].edition << setw(8) << book[num].price;

cout << setw(24) << book[num].ISBN << setw(15) << book[num].numOfEdition << setw(8) << book[num].numOfCopies;

cout << setw(10) << book[num].location << book[num].status << endl;

}

return;

}

// search function

void search(BOOK book[], int size)

{

int method;

cout << "Please choose a method for searching the books record.\nSearch by:\n1 - Title\n2 - Author\n3 - ISBN\n";

cin >> method; // allow user to choose the method to search book

int numOfMatch = 0;

switch (method) {

case 1:

char title[50];

cout << "Please enter a keyword of the title in the book record: ";

cin >> title;

cout << "\nMatched title list:\n";

cout << "No.\tTitle\t\t\t Author\t\t Publisher/Imprint\tEdition Price ISBN\t\t No.of Edition No.of Copies Location Status\n";

for (int num = 0; num < size; num++) {

if (strstr(book[num].title, title) != NULL) {

numOfMatch++; // count for number of matched result

cout << numOfMatch << '\t'; // list out all matched result

cout << left << setw(25) << book[num].title << setw(17) << book[num].author;

cout << setw(22) << book[num].publisher << setw(10) << book[num].edition << setw(8) << book[num].price;

cout << setw(24) << book[num].ISBN << setw(15) << book[num].numOfEdition << setw(8) << book[num].numOfCopies;

cout << setw(10) << book[num].location << book[num].status << endl;

}

}

if (numOfMatch == 0) {

cout << "\nNo matched title record found.\n";

}

break;

case 2:

char author[50];

cout << "Please enter a keyword of author in the book record: ";

cin >> author;

cout << "\nMatched author list:\n";

cout << "No.\tTitle\t\t\t Author\t\t Publisher/Imprint\tEdition Price ISBN\t\t No.of Edition No.of Copies Location Status\n";

for (int num = 0; num < size; num++) {

if (strstr(book[num].author, author) != NULL) {

numOfMatch++; // count for number of matched result

cout << numOfMatch << '\t';// list out all matched result

cout << left << setw(25) << book[num].title << setw(17) << book[num].author;

cout << setw(22) << book[num].publisher << setw(10) << book[num].edition << setw(8) << book[num].price;

cout << setw(24) << book[num].ISBN << setw(15) << book[num].numOfEdition << setw(8) << book[num].numOfCopies;

cout << setw(10) << book[num].location << book[num].status << endl;

}

}

if (numOfMatch == 0) {

cout << "\nNo matched author record found.\n";

}

break;

case 3:

char ISBN[50];

cout << "Please enter a keyword of the ISBN in the book record: ";

cin >> ISBN;

cout << "\nMatched ISBN list:\n";

cout << "No.\tTitle\t\t\t Author\t\t Publisher/Imprint\tEdition Price ISBN\t\t No.of Edition No.of Copies Location Status\n";

for (int num = 0; num < size; num++) {

if (strstr(book[num].ISBN, ISBN) != NULL) {

numOfMatch++; // count for number of matched result

cout << numOfMatch << '\t';// list out all matched result

cout << left << setw(25) << book[num].title << setw(17) << book[num].author;

cout << setw(22) << book[num].publisher << setw(10) << book[num].edition << setw(8) << book[num].price;

cout << setw(24) << book[num].ISBN << setw(15) << book[num].numOfEdition << setw(8) << book[num].numOfCopies;

cout << setw(10) << book[num].location << book[num].status << endl;

}

}

if (numOfMatch == 0) {

cout << "\nNo matched ISBN record found.\n";

}

break;

default:

cout << "Please enter only number 1, 2 or 3 for your choice." << endl;

}

return;

}

// delete record function

void deleteRecord(BOOK book[], int size)

{

list(book, size); // list out available record for deletetion

int bookNumForDel;

int bookNum = 1;

cout << "\nPlease enter the number of respective book record you would like to delete: ";

cin >> bookNumForDel;

ofstream tempFile("temp.txt");

if (!tempFile) {

cout << "Error opening file.\n"; // test if file open successfully

exit(100);

}

else {

if (bookNumForDel != 1) { // test for first record then write it in temp file

tempFile << book[0].title << '\n' << book[0].author << '\n' << book[0].publisher;

tempFile << '\n' << book[0].edition << '\n' << book[0].price << '\n' << book[0].ISBN;

tempFile << '\n' << book[0].numOfEdition << '\n' << book[0].numOfCopies;

tempFile << '\n' << book[0].location << '\n' << book[0].status;

}

for (int num = 1; num < size; num++) {

bookNum++;

if (bookNumForDel != bookNum) { // test for remaining record and write it in temp file

tempFile << "\n\n" << book[num].title << '\n' << book[num].author << '\n' << book[num].publisher;

tempFile << '\n' << book[num].edition << '\n' << book[num].price << '\n' << book[num].ISBN;

tempFile << '\n' << book[num].numOfEdition << '\n' << book[num].numOfCopies;

tempFile << '\n' << book[num].location << '\n' << book[num].status;

}

}

}

tempFile.close(); // close file

remove("booksList.txt"); // delete original file

rename("temp.txt", "booksList.txt"); // rename temp file to original file

cout << "The following book record has been deleted successfully.\n";

return;

}

1984

George Orwell

ChongWenGuan,2006

2006

RM20

PR6029.R8N56123

1

3

Kampar

Available

Metamorphosis

Franz Kafka

ClassixPress,2009

2009

RM15

PT2621.A26V4713

1

2

Sg.Long

Available

Pride and Prejudice

Jane Austen

RenMinChubanshe,2003

2003

RM30

PR4034.P75C5

1

3

Sg.Long

Available

Animal Farm

George Orwell

ClassixPress,2001

2001

RM10

PR6029.R8A56

1

2

Sg.Long

Available

Memoirs of a Geisha

Arthur Golden

Vintage, 2005

2005

RM30

PS3557.O35926M46

1

1

Kampar

Available

The Great Gastby

Scott Fritz

Scribner,1995

1995

RM15

PS3511.I9G74

1

3

Kampar

Available

To Kill a Mockingbird

Harper Lee

WarnerBook,1960

1960

RM10

PZ4.L4778L44

1

2

Sg.Long

Available

Catcher in the Rye

Jerome David

Brown,1951

1951

RM20

PS3557.A426C38

1

3

Kampar

Available

A Storm of Swords

George Martin

BantamBooks,2011

2001

RM30

PS3563.A7239S76

1

4

Sg.Long

Available

A Clash of Kings

George Martin

BantamBooks,2011

2011

RM30

PS3563.A7239C53

1

2

Kampar

Available

The Hobbit

Tolkien

HarperCollins,2011

1999

RM20

PR6039.O32H63

1

1

Sg.Long

Available

Lord of the Rings

Tolkien

HarperCollins,2002

2002

RM20

PR6039.O32L67

1

2

Sg.Long

Available

Moby Dick

Melville Herman

NobleClassics2003

2003

RM25

PS2384.M6

1

2

Sg.Long

Available

Lord of the Flies

William Golding

FABER,1962

1962

RM20

PR6013.O35L67

1

3

Kampar

Available

Brave New World

Aldous Huxley

HarperRow,1965

1963

RM15

PR6015.U9B73

1

1

Sg.Long

Available

Heart of Darkness

Joseph Conrad

RenMinChubanshe,1998

2006

RM10

PR6005.O4L67123

1

2

Kampar

Available

Oliver Twist

Charles Dicken

PenguinBooks,1994

1998

RM15

PR4567.A1O45

1

3

Kampar

Available

Treasure Island

Robert Louis

Thomson,2006

2006

RM25

PR5486.A1

1

3

Sg.Long

Available

Persuasion

Jane Austen

PenguinBooks,1994

1994

RM20

PR4034.P47

1

2

Sg.Long

Available

Robinson Crusoe

Daniel Defoe

Oxford, 2007

2007

RM25

PR3403.A2K49

1

2

Kampar

Available